Lecture Notes in Computer Science 4238

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Young-Tak Kim Makoto Takano (Eds.)

Management of Convergence Networks and Services

9th Asia-Pacific Network Operations
and Management Symposium, APNOMS 2006
Busan, Korea, September 27-29, 2006
Proceedings

Springer

Volume Editors

Young-Tak Kim
Yeungnam University
School of Electronics Engineering and Computer Science
214-1, Dae-Dong, Kyungsan-Si, Kyungbook, 721-749, Korea
E-mail: ytkim@yu.ac.kr

Makoto Takano
NTT West R&D Center
IT Governance Promoting Group
6-2-82, Shimaya, Konohana-ku, Osaka, 554-0024, Japan
E-mail: m.takano@rdc.west.ntt.co.jp

Library of Congress Control Number: 2006932881

CR Subject Classification (1998): C.2, D.2, D.4.4, K.6, H.3-4

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

ISSN 0302-9743
ISBN-10 3-540-45776-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-45776-3 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11876601 06/3142 5 4 3 2 1 0

Preface

We are delighted to present the proceedings of the 9th Asia-Pacific Network Operations and Management Symposium (APNOM S2006) which was held in Busan, Korea, on September 27-29, 2006.

Recently, various convergences in wired and wireless networks and convergence of telecommunications and broadcastings are taking place for ubiquitous multimedia service provisioning. For example, broadband IP/MPLS wired networks are actively converged with IEEE 802.11e wireless LAN, IEEE 802.16 Wireless MAN, 3G/4G wireless cellular networks, and direct multimedia broadcast (DMB) networks. For efficient support of service provisioning for ubiquitous multimedia services on the broadband convergence networks, well-designed and implemented network operations and management functions with QoS-guaranteed traffic engineering are essential. The Organizing Committee (OC) made the timely selection of "Management of Convergence Networks and Services" as the theme of APNOMS 2006.

Contributions from academia, industry and research institutions met these challenges with 280 paper submissions, from which 50 high-quality papers were selected for technical sessions as full papers, and 25 papers as short papers. The diverse topics in this year's program included management of ad-hoc and sensor networks, network measurements and monitoring, mobility management, QoS management, management architectures and models, security management, E2E (end-to-end) QoS and application management, management experience, NGN (next-generation network) management, and IP-based network management.

The Technical Program Committee (TPC) Co-chairs would like to thank all those authors who contributed to the outstanding APNOMS 2006 technical program, and thank the TPC and OC members and reviewers for their support throughout the paper review and program development process. Also, we appreciate KICS KNOM, Korea, and IEICE TM, Japan, for their sponsorship, as well as IEEE CNOM, IEEE APB, TMF and IFIP WG 6.6 for their support for APNOMS 2006.

September 2006

Young-Tak Kim
Makoto Takano

Organizing Committee

General Chair

James Hong, POSTECH, Korea

Vice Chair

Hiroshi Kuriyama, NEC, Japan

TPC Co-chairs

Young-Tak Kim, Yeungnam University, Korea
Makoto Takano, NTT West, Japan

Tutorial Co-chairs

Dongsik Yun, KT, Korea
Toshio Tonouchi, NEC, Japan

Special Session Co-chairs

Jae-Hyoung Yoo, KT, Korea
Kazumitsu Maki, Fujitsu, Japan

Keynotes Chair

Doug Zuckerman, Telcordia, USA

DEP Chair

G. S. Kuo, National Chengchi University, Taiwan

Exhibition Co-chairs

Young-Woo Lee, KT, Korea
Takashi Futaki, Microsoft, Japan

Poster Co-chairs

Wang-Cheol Song, Cheju National University, Korea
Shingo Ata, Osaka City University, Japan

Publicity Co-chairs

Gilhaeng Lee, ETRI, Korea
Choong Seon Hong, KHU, Korea
Qinzheng Kong, HP APJ, Australia
Kiminori Sugauchi, Hitachi, Japan

Finance Co-chairs

Hong-Taek Ju, Keimyung University, Korea
Kohei Iseda, Fujitsu Labs., Japan

Publication Chair

YoungJoon Lee, KNUE, Korea

Local Arrangements Co-chairs

Kwang-Hui Lee, Changwon University, Korea
Jae-Min Eom, KTF, Korea

Secretaries

Jae-Oh Lee, KUT, Korea
Hideo Imanaka, NTT, Japan

International Liaisons

Takeo Hamada, Fujitsu Labs of America, USA
Raouf Boutaba, University of Waterloo, Canada
Carlos Westphall, Santa Catalina Federal University, Brazil
Hiroyuki Okazaki, NEC Europe Ltd., Germany
Rajan Shankaran, Macquarie University, Australia
Alpna J. Doshi, Satyam Computer Services, India
Teerapat Sanguankotchakorn, AIT, Thailand
Ryoichi Komiya, Multimedia University, Malaysia
Victor WJ Chiu, Chunghwa Telecom, Taiwan
Yan Ma, Beijing University of Posts and Telecommunications, China

Advisory Board

Young-Hyun Cho, KTH, Korea
Young-Myoung Kim, KT, Korea
Graham Chen, EPAC Tech, Australia
Makoto Yoshida, University of Tokyo, Japan
Masayoshi Ejiri, Fujitsu, Japan
Doug Zuckerman, Telcordia, USA

Standing Committee

Nobuo Fuji, NTT, Japan
Hiroshi Kuriyama, NEC, Japan
James W. Hong, POSTECH, Korea
Kyung-Hyu Lee, ETRI, Korea
Seong-Beom Kim, KT, Korea
Yoshiaki Tanaka, Waseda University, Japan

Technical Program Committee

Nazim Agoulmine, University of Evry, France
Joseph Betser, The Aerospace Corporation, USA
Raouf Boutaba, University of Waterloo, Canada
Marcus Brunner, NEC Europe, Germany
Prosper Chemouil, France Telecom, France
Graham Chen, CiTR, Australia
Taesang Choi, ETRI, Korea
Young-Bae Choi, James Madison University, USA
Idir Fodil, France Telecom, France
Lisandro Granville, FURGS, Brazil
Takeo Hamada, Fujitsu Labs of America, USA
Choong Seon Hong, Kyung Hee University, Korea
Gabriel Jakobson, Altusys, USA
Gautam Kar, IBM Research, USA
Yoshiaki Kiriha, NEC, Japan
Kwang-Hui Lee, Changwon University, Korea
Alberto Leon-Garcia, University of Toronto, Canada
Antonio Liotta, University of Essex, UK
Souhei Majima, NTT, Japan
Takeshi Masuda, NTT, Japan
Jose Marcos Nogueira, FUMG, Brazil
Takao Ogura, Fujitsu Lab, Japan
Sungjin Ahn, Sungkyunkwan University, Korea
Aiko Pras, University of Twente, Netherlands
Pradeep Ray, UNSW, Australia
Teerapat Sa-nguankotchakorn, AIT, Thailand
Akhil Sahai, HP Labs, USA
Joan Serrat, UPC, Spain
Rajan Shankaran, Macquarie University, Australia
Haizhou Shi, HP, China
Radu State, INRIA, France
John Strassner, Motorola, USA
Memhet Ulema, Manhattan College, USA
Carlos Westphal, FUSC, Brazil
Felix Wu, UC Davis, USA
Jae-Hyung Yoo, KT, Korea

Jianqiu Zeng, BUPT, China
Yongbing Zhang, University of Tsukuba, Japan
Doug Zuckerman, Telcordia, USA

Additional Reviewers

Daniel W. Hong, KT, Korea
Deok-Jae Choi, Chonnam University, Korea
Hiroki Horiuchi, KDDI R&D Labs., Japan
Jae-Il Chung, Hanyang University, Korea
Jaehwoon Lee, Dongguk University, Korea
Jong-Tae Park, KNU, Korea
Jun Hwang, SNU, Korea
Katsushi Iwashita, Kochi Tech., Japan
Ki-Hyung Kim, Ajou University, Korea
Myung-Kyun Kim, University of Ulsan, Korea
Naoto Miyauchi, Mitsubishi Electric, Japan
Satoru Sugimoto, ANSL NTT, Japan
Seong-Bae Eun, University of Hannam, Korea
Si-Ho Cha, Sejong University, Korea
Takaaki Hasegawa, NTT West, Japan
Tetsuya Yamamura, NTT, Japan
Toshio Nishiyama, NTT, Japan
Wan-Sup Cho, CNU, Korea
Yoon-Hee Kim, Sookmyung Women's University, Korea
Youichi Yamashita, NTT, Japan
Youngsu Chae, Yeungnam University, Korea

Table of Contents

Session 1: Management of Ad-Hoc and SensorNetworks

QoS-Aware Fair Scheduling in Wireless Ad Hoc Networks with Link Errors 1
Muhammad Mahbub Alam, Md. Mamun-or-Rashid, Choong Seon Hong

Performance Analysis of Service Differentiation for IEEE 802.15.4 Slotted CSMA/CA 11
Meejoung Kim

Information-Driven Task Routing for Network Management in Wireless Sensor Networks 23
Yu Liu, Yumei Wang, Lin Zhang, Chan-hyun Youn

Autonomic Management of Scalable Load-Balancing for Ubiquitous Networks 33
Toshio Tonouchi, Yasuyuki Beppu

A Policy-Based Management Framework for Self-managed Wireless Sensor Networks 43
Jong-Eon Lee, Si-Ho Cha, Jae-Oh Lee, Seok-Joong Kang, Kuk-Hyun Cho

Session 2: Network Measurements and Monitoring

A Proposal of Large-Scale Traffic Monitoring System Using Flow Concentrators 53
Atsushi Kobayashi, Daisuke Matsubara, Shingo Kimura, Motoyuki Saitou, Yutaka Hirokawa, Hitoaki Sakamoto, Keisuke Ishibashi, Kimihiro Yamamoto

Novel Traffic Measurement Methodology for High Precision Applications Awareness in Multi-gigabit Networks 63
Taesang Choi, Sangsik Yoon, Dongwon Kang, Sangwan Kim, Joonkyung Lee, Kyeongho Lee

Rate-Based and Gap-Based Available Bandwidth Estimation Techniques in Cross-Traffic Context 73
Wayman Tan, Marat Zhanikeev, Yoshiaki Tanaka

Signature-Aware Traffic Monitoring with IPFIX 82
Youngseok Lee, Seongho Shin, Taeck-geun Kwon

Temporal Patterns and Properties in Multiple-Flow Interactions 92
Marat Zhanikeev, Yoshiaki Tanaka

Session 3: Mobility Management

A Profile Based Vertical Handoff Scheme for Ubiquitous Computing Environment 102
Chung-Pyo Hong, Tae-Hoon Kang, Shin-Dug Kim

The Soft Bound Admission Control Algorithm for Vertical Handover in Ubiquitous Environment 112
Ok Sik Yang, Jong Min Lee, Jun Kyun Choi, Seong Gon Choi, Byung Chun Jeon

Improving Handoff Performance by Using Distance-Based Dynamic Hysteresis Value 122
Huamin Zhu, Kyungsup Kwak

A Seamless Service Management with Context-Aware Handoff Scheme in Ubiquitous Computing Environment 132
Tae-Hoon Kang, Chung-Pyo Hong, Won-Joo Jang, Shin-Dug Kim

Performance Analysis of an Adaptive Soft Handoff Algorithm for Mobile Cellular Systems 142
Huamin Zhu, Kyungsup Kwak

Session 4: QoS Management

An Admission Control and Traffic Engineering Model for Diffserv-MPLS Networks 152
Haci A. Mantar

An Admission Control and TXOP Duration of VBR Traffics in IEEE 802.11e HCCA with Guaranteed Delay and Loss 162
Tae Ok Kim, Yong Chang, Young-Tak Kim, Bong Dae Choi

NETSAQ: Network State Adaptive QoS Provisioning for MANETs 170
Shafique Ahmad Chaudhry, Faysal Adeem Siddiqui, Ali Hammad Akbar, Ki-Hyung Kim

End-to-End QoS Guaranteed Service in WLAN and 3GPP Interworking Network 180
Sung-Min Oh, Jae-Hyun Kim, You-Sun Hwang, Hye-Yeon Kwon, Ae-Soon Park

Network-Adaptive QoS Routing Using Local Information 190
Jeongsoo Han

Session 5: Management Architectures and Models

Configuration Management Policy in QoS-Constrained Grid Networks ... 200
Hyewon Song, Chan-Hyun Youn, Chang-Hee Han, Youngjoo Han, San-Jin Jeong, Jae-Hoon Nah

A Proposal of Requirement Definition Method with Patterns for Element / Network Management 210
Masataka Sato, Masateru Inoue, Takashi Inoue, Tetsuya Yamamura

Distributed Fault Management in WBEM-Based Inter-AS TE for QoS Guaranteed DiffServ-over–MPLS 221
Abdurakhmon Abdurakhmanov, Shahnaza Tursunova, Shanmugham Sundaram, Young-Tak Kim

A Framework Supporting Quality of Service for SOA-Based Applications 232
Phung Huu Phu, Dae Seung Yoo, Myeongjae Yi

Performance Improvement Methods for NETCONF-Based Configuration Management 242
Sun-Mi Yoo, Hong Taek Ju, James Won-Ki Hong

Session 6: Security Management

Zone-Based Clustering for Intrusion Detection Architecture in Ad-Hoc Networks 253
Il-Yong Kim, Yoo-Sung Kim, Ki-Chang Kim

Tracing the True Source of an IPv6 Datagram Using Policy Based Management System 263
Syed Obaid Amin, Choong Seon Hong, Ki Young Kim

An Efficient Authentication and Simplified Certificate Status Management for Personal Area Networks 273
Chul Sur, Kyung Hyune Rhee

A Novel Rekey Management Scheme in Digital Broadcasting Network ... 283
Han-Seung Koo, Il-Kyoo Lee, Jae-Myung Kim, Sung-Woong Ra

A New Encoding Approach Realizing High Security and High Performance Based on Double Common Encryption Using Static Keys and Dynamic Keys ... 293
Kiyoshi Yanagimoto, Takaaki Hasegawa, Makoto Takano

Session 7: E2E QoS and Application Management

GMPLS-Based VPN Service to Realize End-to-End QoS and Resilient Paths ... 302
Hiroshi Matsuura, Kazumasa Takami

WBEM-Based SLA Management Across Multi-domain Networks for QoS-Guaranteed DiffServ-over-MPLS Provisioning ... 312
Jong-Cheol Seo, Hyung-Soo Kim, Dong-Sik Yun, Young-Tak Kim

Network Support for TCP Version Migration ... 322
Shingo Ata, Koichi Nagai, Ikuo Oka

End-to-End QoS Monitoring Tool Development and Performance Analysis for NGN ... 332
ChinChol Kim, SangChul Shin, SangYong Ha, SunYoung Han, YoungJae Kim

"P4L": A Four Layers P2P Model for Optimizing Resources Discovery and Localization ... 342
Mourad Amad, Ahmed Meddahi

Session 8: Management Experience

A Zeroconf Approach to Secure and Easy-to-Use Remote Access to Networked Appliances ... 352
Kiyohito Yoshihara, Toru Maruta, Hiroki Horiuchi

A Test Method for Base Before Service (BS) of Customer Problems for the NeOSS System ... 362
InSeok Hwang, SeungHak Seok, JaeHyoong Yoo

Self-management System Based on Self-healing Mechanism ... 372
Jeongmin Park, Giljong Yoo, Chulho Jeong, Eunseok Lee

Experiences in End-to-End Performance Monitoring on KOREN ... 383
Wang-Cheol Song, Deok-Jae Choi

SOA-Based Next Generation OSS Architecture 393
Young-Wook Woo, Daniel W. Hong, Seong-Il Kim, Byung-Soo Chang

Session 9: NGN Management

Performance Analysis of a Centralized Resource Allocation Mechanism for Time-Slotted OBS Networks 403
Tai-Won Um, Jun Kyun Choi, Seong Gon Choi, Won Ryu

Efficient Performance Management of Subcarrier-Allocation Systems in Orthogonal Frequency-Division Multiple Access Networks 412
Jui-Chi Chen, Wen-Shyen E. Chen

Convergence Services Through NGN-CTE on the Multiple Service Provider Environments in NGN 422
Soong Hee Lee, Haeng Suk Oh, Dong Il Kim, Hee Chang Chung, Jong Hyup Lee

Proposal of Operation Method for Application Servers on NGN Using Unified Management Environment 431
Atsushi Yoshida, Yu Miyoshi, Yoshihiro Otsuka

IP/WDM Optical Network Testbed: Design and Implementation 441
H.A.F. Crispim, Eduardo T.L. Pastor, Anderson C.A. Nascimento, H. Abdalla Jr , A.J.M. Soares

Session 10: IP-Based Network Management

Voice Quality Management for IP Networks Based on Automatic Change Detection of Monitoring Data 451
Satoshi Imai, Akiko Yamada, Hitoshi Ueno, Koji Nakamichi, Akira Chugo

Parameter Design for Diffusion-Type Autonomous Decentralized Flow Control 461
Chisa Takano, Keita Sugiyama, Masaki Aida

Bandwidth Management for Smooth Playback of Video Streaming Services 471
Hoon Lee, Yoon Kee Kim, Kwang-Hui Lee

An Enhanced RED-Based Scheme for Differentiated Loss Guarantees 481
Jahwan Koo, Vladimir V. Shakhov, Hyunseung Choo

Dynamic Location Management Scheme Using Agent in a Ubiquitous IP-Based Network 491
Soo-Young Shin, Soo-Hyun Park, Byeong-Hwa Jung, Chang-Hwa Kim

Short Paper Session

Detecting and Identifying Network Anomalies by Component Analysis 501
Le The Quyen, Marat Zhanikeev, Yoshiaki Tanaka

Preventive Congestion Control Mechanisms in ATM Based MPLS on BcN: Detection and Control Mechanisms for a Slim Chance of Label Switched Path 505
Chulsoo Kim, Taewan Kim, Jin hyuk Son, Sang Ho Ahn

Scalable DiffServ-over-MPLS Traffic Engineering with Per-flow Traffic Policing 509
Djakhongir Siradjev, Ivan Gurin, Young-Tak Kim

On the Dynamic Management of Information in Ubiquitous Systems Using Evolvable Software Components 513
Syed Shariyar Murtaza, Bilal Ahmed, Choong Seon Hong

A Shared-Memory Packet Buffer Management in a Network Interface Card 517
Amit Uppal, Yul Chu

An Adaptive Online Network Management Algorithm for QoS Sensitive Multimedia Services 521
Sungwook Kim, Sungchun Kim

Improved Handoff Performance Based on Pre-binding Update in HMIPv6 525
Jongpil Jeong, Min Young Chung, Hyunseung Choo

On the Security of Attribute Certificate Structuring for Highly Distributed Computing Environments 530
Soomi Yang

Performance Analysis of Single Rate Two Level Traffic Conditioner for VoIP Service 534
Dae Ho Kim, Ki Jong Koo, Tae Gyu Kang, Do Young Kim

An Architectural Framework for Network Convergence Through Application Level Presence Signaling 538
Atanu Mukherjee

Security Approaches for Cluster Interconnection in a Wireless Sensor Network 542
Alexandre Gava Menezes, Carlos Becker Westphall

A Resource-Optimal Key Pre-distribution Scheme with Enhanced Security for Wireless Sensor Networks 546
Tran Thanh Dai, Al-Sakib Khan Pathan, Choong Seon Hong

Intelligent Home Network Service Management Platform Design Based on OSGi Framework 550
Choon-Gul Park, Jae-Hyoung Yoo, Seung-Hak Seok, Ju-Hee Park, Hoen-In Lim

COPS-Based Dynamic QoS Support for SIP Applications in DSL Networks 554
Seungchul Park, Yanghee Choi

IP Traceback Algorithm for DoS/DDoS Attack 558
Hong-bin Yim, Jae-il Jung

An Open Service Platform at Network Edge 562
Dong-Hui Kim, Jae-Oh Lee

Hybrid Inference Architecture and Model for Self-healing System 566
Giljong Yoo, Jeongmin Park, Eunseok Lee

A Node Management Tool for Dynamic Reconfiguration of Application Modules in Sensor Networks 570
Sunwoo Jung, Jaehyun Choi, Dongkyu Kim, Kiwon Chong

Path Hopping Based on Reverse AODV for Security 574
Elmurod Talipov, Donxue Jin, Jaeyoun Jung, Ilkhyu Ha, YoungJun Choi, Chonggun Kim

Mixing Heterogeneous Address Spaces in a Single Edge Network 578
Il Hwan Kim, Heon Young Yeom

Delivery and Storage Architecture for Sensed Information Using SNMP 582
DeokJai Choi, Hongseok Jang, Kugsang Jeong, Punghyeok Kim, Soohyung Kim

GISness System for Fast TSP Solving and Supporting Decision Making 586
Iwona Pozniak-Koszalka, Ireneusz Kulaga, Leszek Koszalka

A DNS Based New Route Optimization Scheme with Fast Neighbor Discovery in Mobile IPv6 Networks 590
Byungjoo Park, Haniph Latchman

Performance Analysis of Group Handoff in Multihop Mesh Relay System 594
Young-uk Chung, Yong-Hoon Choi, Hyukjoon Lee

DSMRouter: A DiffServ-Based Multicast Router 598
Yong Jiang

Author Index 603

QoS-Aware Fair Scheduling in Wireless Ad Hoc Networks with Link Errors*

Muhammad Mahbub Alam, Md. Mamun-or-Rashid, and Choong Seon Hong**

Department of Computer Engineering, Kyung Hee University
1 Seocheon, Giheung, Yongin, Gyeonggi, Korea, 449-701
{mahbub, mamun}@networking.khu.ac.kr,
cshong@khu.ac.kr

Abstract. To provide scheduling in wireless ad hoc networks, that is both highly efficient and fair in resource allocation, is not a trivial task because of the unique problems in wireless networks such as location dependent and bursty errors in wireless link. A packet flow in such a network may be unsuccessful if it experiences errors. This may lead to situations in which a flow receives significantly less service than it is supposed to, while other receives more, making it difficult to provide fairness. In this paper we propose a QoS-aware fair scheduling mechanism in ad hoc networks considering guaranteed and best-effort flows in the presence of link errors. The proposed mechanism provides short-term fairness for error free sessions and long-term fairness for the erroneous sessions and allows a lagging flow to receive extra service and a leading flow to give up its extra service in a graceful way. It also maximizes the resource utilization by allowing spatial reuse of resource. We also propose a CSMA/CA based implementation of our proposed method.

1 Introduction

A wireless ad hoc network consists of a group of mobile nodes without the support of any infrastructure. Such a network is expected to support advanced applications such as communications in emergency disaster management, video conferencing in a workshop or seminar, communications in a battlefield. This class of mission-critical applications demands a certain level of quality of services (QoS) for proper operations. Also due to the distributed nature of these networks providing a fair access of resource to multiple contending nodes is an important design issue.

Fairness is an important criterion of resource sharing in the best effort Internet, especially when there is a competition for the resource among the nodes due to unsatisfied demands. In fair scheduling each flow f is allowed to share a certain percentage of link capacity based on its flow weight indicated as w_f. Let $W_f(t1, t2)$ and $W_g(t1, t2)$ denote the aggregate resources received by flows f and g respectively in time interval $[t1, t2]$ and w_f and w_g are the flow weights of the flows f and g respectively. The allocation is ideally fair if it satisfies (1)

* This work was supported by MIC and ITRC Project.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 1 – 10, 2006.
© Springer-Verlag Berlin Heidelberg 2006

$$\left| \frac{W_f(t_1, t_2)}{w_f} - \frac{W_g(t_1, t_2)}{w_g} \right| = 0 \quad (1)$$

However, most of the research works assumed error free wireless link which is not realistic. In the wireless environment, a packet flow may experience channel error and the transmission may not be successful. Thus the bursty and location dependent error in wireless link may make the existing fair scheduling algorithm inapplicable. Therefore, the goal of ad hoc network fair scheduling is to make short burst of location-dependent channel error transparent to users by a dynamic reassignment of channel allocation over small timescales [14]. Specifically, a backlogged flow *f* that perceives a channel error during a time window $[t_1, t_2]$ is compensated over a later time window $[t_1', t_2']$ when *f* perceives a clean channel. Compensation for *f* involves granting additional channel access to *f* during $[t_1', t_2']$ in order to make up lost channel access during $[t_1, t_2]$, and this additional channel access is granted to *f* at the expense of flows that were granted additional channel access during $[t_1, t_2]$.

Providing QoS in Wireless ad hoc networks is a new area of research. Existing works focuse mainly on QoS routing which finds a path to meet the desired service requirements of a flow. In this paper we consider a mix of guaranteed and best effort flows and investigate fair queueing with QoS support for the network. The goal is to guarantee the minimum bandwidth requirements of guaranteed flows and to ensure a fair share of residual bandwidth to all flows.

In this paper we focus on the fair scheduling issues in a hoc network in the presence of channel errors. We develop a fairness model for wireless ad hoc fair scheduling to deal with channel error. We also implement the model in a distributed manner by localizing the global information required by the nodes.

The rest of the paper is organized as follows. Section 2 describes related works. In Section 3 we explain the network model and problem specifications. Section 4 describes the proposed mechanism and is followed by the details of the implementation of the proposed mechanism in section 5. Section 6 presents the simulation and results. We conclude in section 7 by conclusion and future works.

2 Related Works

Fair queueing has been a popular paradigm for providing fairness, minimum throughput assurance and guaranteed delay in wired network [1], and in packet cellular networks [2] – [4]. Recently some techniques have been proposed to incorporated fair queueing in shared channel, multihop wireless networks [5] – [7]. Also, providing QoS in wireless ad hoc networks is a new area of research. Some of the research works also incorporated both QoS and fair queueing in ad hoc networks. Both QoS guarantee and fair queueing in ad hoc networks have been proposed in [8] and [9]. Also some of the works provide fairness with error compensation, e.g., [13], [14], [15] and most of these are proposed based on the support of base stations.

In [13], channel-condition independent packet fair queueing (CIF-Q) is proposed. Each session is associated with a parameter called *lag* to indicate whether the session should be compensated. If a session is not leading and the channel is error free at its scheduled time, its head-of-line packet is transmitted; otherwise, the time slot is

released to other sessions. The problem with CIF-Q is that the leading sessions are not allowed to terminate unless all their leads have been paid back, regardless of whether such terminations are caused by broken routes. This property makes CIF-Q an inadequate solution for ad hoc networks, because a connection may be broken.

The Idealized Wireless Fair-Queueing (IWFQ), and the Wireless Packet Scheduling protocol (WPS) are proposed in [14]. In this paper the base station calculates the number of slots per frame, each flow can use, based on the weight of the flows. If a flow experiences channel error at its allocated time, the base station tries to find a flow that can exchange its slot with the flow within the frame; and the error session of the flow is compensated at a later frame. The mechanism is not suitable for ad hoc networks, as it requires the presence of a base station.

In [15], QoS-aware fair scheduling is proposed for mobile ad hoc networks in the presence of error; and based on the channel condition estimation, a flow may either transmit or give up its allocation to others. When this flow perceives an error free channel it will have packets with smaller service tag than that of its neighbors' and its packets will be transmitted first. The problem with this protocol is that after recovery from error a flow exclusively accesses the channel until its virtual time catches up with other flows. Also the service release of a leading flow is not graceful.

3 Network Model and Problem Specifications

In this paper we consider shared channel, packet-switched, multihop wireless ad hoc networks where the hosts are not mobile. There is contention for channel among multiple nodes and errors in wireless link are location dependent and bursty in nature. Therefore, some hosts may not be able to transmit data due to channel errors even when there is backlogged flows on those hosts while others may have error-free channels and can transmit data in that time.

We define the error-free service of a flow as the service that had been error-free. A flow is said to be leading if it has received channel allocation in excess of its error-free service. A flow is said to be lagging if it has received channel allocation less than its error-free service. If a flow is neither leading nor lagging, it is said to be in sync.

During a period of channel error, error-free flows will receive more service than their fair share, while a flow with errors will receive no service. Since the virtual time of a session increases when it receives service, this may result in a large difference between the virtual time of an erroneous flow and that of an error-free session. We address the following issues of ad hoc network fair scheduling with channel errors:

i) If flow f_i exits from errors, and is allowed to retain its virtual time, then it will have the smallest virtual time among all flows. Then f_i will exclusively access the channel until its virtual time catches up with those of other flows and all the other flows will receive no service.

ii) If flow f_i exits from error, and its virtual time is updated to the system virtual time $V(\bullet)$, then error-free flows will not be penalized and f_i will never be able to regain its lost service, resulting in unfair behaviors.

iii) The compensation of all lagging flows should not take same amount of time regardless of their weights, and violate the main idea that larger weight implies better service.

iv) The extra service releases by a leading flow should be graceful to guarantee short-term fairness.

4 Proposed Fair Scheduling Mechanism

We now describe a new approach to QoS-aware distributed fair scheduling model in ad hoc networks. This model is fully distributed, localized and local scheduler at each node has a certain level of coordination with its neighboring nodes. And this does not require any global information propagation and global computation. The mechanism is as follows:

1) Maintaining Flow Information within Two-hop Neighborhood: Each node maintains two tables for the proper operation of fair scheduling. One table is to keep the flow information of two-hop neighbors, say *flow_table* and is kept sorted according to service tag of the flows. The fields of the table are *node_id*, *flow_id*, *service_tag,* s_i, and *lag*. Another table, *compensation_table*, contains the compensation tag of the lagging flows. The fields of the table are *node_id*, *flow_id* and *compensation_tag* and kept sorted according to the *compensation_tag*.

2) Assignment of flow weight: We assume that both guaranteed and best-effort flows exist simultaneously in the network. Flow weights are assigned based on [16]:

Weight of QoS flows, w_q	Weight of best-effort flows, w_b

For i = 1 to n {
w =Min_f/C +(C-$\sum Min_f$)/(n + m)
if w*C > Req_f
W_q = C/Req_f
else
W_q = w
}
	For i = 1 to m {
	w_b =(C – ($\sum w_g$)*C)/m
	}

m = number of QoS flows C = Link Bandwidth	m = number of best effort flows

We consider a multi-hop flow consists of number of single hop flows and the flow weight is not fixed for the path; every forwarding node assigns a different weight. Over time, based on the current flows within two-hop neighbors, the weight may change, otherwise either the network utilization or the fairness will be poor.

3) Tagging Operations: For each flow *f* we use the SFQ [10] algorithm to assign tags for the arriving packets: a *start tag* and a *finish tag*.

4) Path Registration: To provide guaranteed service a routing protocol should find a feasible path. AODV [11] is one of the most widely used table-based and reactive routing protocols. But AODV is designed to find a feasible route only. Therefore, to support QoS we need to modify AODV. To ensure QoS, AODV is modified to support two types of schemes: *admission scheme* and *adaptive scheme*. In admission scheme a feasible path should provide the required minimum bandwidth, while in adaptive feedback scheme the source is informed about the minimum available bandwidth so that the source can adapt its transmission speed.

To initiate QoS-aware routing discovery, the source host sends a RREQ packet whose header is changed to <*model-flag*, *required bandwidth*, *min-bandwidth*, AODV RREQ header>. The model-flag indicates whether the source is using the admission scheme or the adaptive feedback scheme. When an intermediate host receives the RREQ packet, it first calculates its residual bandwidth. If the model-flag is set to admission scheme, the host compares its residual bandwidth with the minimum requested bandwidth. If its residual bandwidth is greater than the minimum bandwidth, it forwards this RREQ. Otherwise, it discards this RREQ. If the model-flag is adaptive, the host compares its residual bandwidth with the min-bandwidth field in the RREQ. If its residual bandwidth is greater than the min-bandwidth, it forwards the RREQ. Otherwise, it updates the min-bandwidth value using its residual bandwidth. Finally the forwarding node temporarily stores the flow. When a node forwards a RREP message it assigns a flow-weight to the flow and stores the flow information.

5) Channel Error Prediction: Perfect channel-dependent scheduling is only possible if a node has accurate information about the channel state. The location-dependent nature of channel error requires each node to monitor its channel state continuously, based on which the node may predict its future channel state. To do this, each node periodically measures the signal-to-noise ratio (SNR) of the channel. If the SNR value falls below a predefined threshold, the channel is considered as error-prone.

6) Lead and Lag Model: The purpose of the lead and lag model is to determine how much additional service a lagging flow is entitled to receive in the future in order to compensate service lost in the past and how much service a leading flow should relinquish in the future in order to give up additional services received in the past. Also, the lag of a lagging flow is incremented on a lost time slot only if another flow received its service and is ready to release this service.

To represent the amount of lead and lag service of a flow we use a parameter called *lag*. The value of *lag* of a lagging flow is positive, it is negative for a leading flow and zero otherwise.

When a node f_i perceives an erroneous wireless link and it is the turn for one of its flow to transmit, the node does not transmit. Instead one of its neighbors, f_j, which perceives a good channel and has the immediate higher service tag (lagging flows get higher preference than leading flows) will transmit. Both of the flows will either initialize (if they just become lagging and leading) or update their *lag* value.

7) Compensation Model: The compensation model is the key component of ad hoc network fair scheduling algorithm in the presence of errors. It determines how lagging flows receive extra service to make up their lag and leading flows give up their lead to relinquish extra service. The compensation model should take into account the following two conditions to ensure short-term fairness:

i) The service releases by a leading flow should not be more than a certain fraction of its received service.

ii) The lagging flows should receive the extra service in proportion to their flow weights. That is, flow with largest *lag* should not receive more service irrespective to its guarantee rate.

To achieve graceful degradation of service of a leading flow, it relinquishes a fraction of services allocated to the flow. We define a system parameter α ($0 \leq \alpha \leq 1$) to control the minimal fraction of service retained by a leading session. That is, a leading flow can give up at most $(1 - \alpha)$ amount of its service to compensate for lagging

flows. Each leading flow, f_i, is associated with another parameter, called normalized service received by the leading flow, s_i. When a flow becomes leading, s_i is initialized to αv_i (v_i is the virtual time of f_i) and s_i is updated whenever f_i is served.

On the other hand, lagging flows should have higher priority to receive extra service to compensate their loss. And such compensation is possible because leading flows give up their leads. To provide short-term fairness we distribute these additional services among lagging flows in proportion to the lagging flows weights. To accomplish this we use the compensation virtual time, c_i, which keeps the track of normalized amount of compensation services received by a flow while it is lagging. When flow f_i becomes lagging c_i is initialized according to (2)

$$c_i = \left\lceil \max(c_i, \min_{k \in A} \{ f_k \mid \mathtt{lag}_k > 0 \} \right\rceil \tag{2}$$

8) Scheduling Mechanism: For scheduling the flows we use a table driven, back-off based approach which uses local information and local computation only. With the tagging operation and a method of exchanging tags, each node has the knowledge of its local neighborhood. These tags are stored in tables and are ordered so that each node can learn whether that node itself has the minimum tag and therefore has to transmit next. The basic scheduling mechanism is as follows:

i) When a node finds one of its flows f_i has the minimum service tag and the flow is not leading or leading but did not receive a minimum fraction of service, then the packet at its queue is transmitted. However, if f_i is leading and already receives minimum service, it does not transmit its packet to give up its lead to a lagging flow. If there is any lagging flow within two-hop neighbors of the flow, then the flow f_j, which has the minimum compensation tag among the lagging flows can transmit its packet, otherwise f_i transmits its packet.

ii) After the packet to transmit has been determined, the virtual time of f_i is updated. If f_i is leading but receives services due to graceful degradation, s_i is updated to $s_i + L_p$. If f_j is served and the overhead is charged to f_i where $(i \neq j)$, then the lag values are update by $lag_j = lag_j - L_p$ and $lag_i = lag_i + L_p$.

iii) When a flow just becomes lagging its *compensation_tag*, c_i is initialized by (2) and updated every time a lagging flow gets compensation service by $c_i = c_i + L_p / w_i$.

When flow f_i becomes leading its s_i is initialized by $s_i = \alpha v_i$ and updated every time it gets service due to graceful degradation by $s_i = s_i + L_p / w_i$.

9) Table Update: Whenever a node hears a new service tag for any flow on its table or a new flow, it updates the table entry for that flow or adds this flow information on its table. Whenever any node transmits a head-of-line packet for a flow, it updates that flow's service tag and compensation tag in the table entry.

5 Implementation of the Proposed Mechanism

In this section, we describe a distributed implementation of the proposed mechanism within the framework of CSMA/CA MAC architecture. Our implementation is based on the method used in [8] and we incorporated the channel errors with this addressing the following practical issues:

1) Message Exchange Sequence: In this mechanism, each data transmission follows a basic sequence of RTS-CTS-DS-DATA-ACK handshake, and this message exchange is preceded by a backoff of certain number of minislot times. At the begging of each transmission slot, each node checks whether it has a flow with minimum service tag. The flow with the minimum service tag is transmitted based on the rules explained next. To allow multiple flows to transmit simultaneously and to reuse the channel we assign a backoff value to flows like [8]. But we assign backoff value first to lagging flows and then to leading flows otherwise the leading flows will further increase their lead. The value of the backoff timer of flow f_i is the number of flows with tag smaller than the tag of the flow f_i.

If the node does not hear any transmission then it decreases backoff value by one in each minislot. If the backoff timer of f_i expires without overhearing any ongoing transmission, it starts RTS to initiate the handshake. If the node overhears some ongoing transmission, it cancels its backoff timer and defers until the ongoing transmission completes. In the meantime, it updates its local tables for the tags of the ongoing neighboring transmitting flow. When other nodes hear a RTS, they defer for one CTS transmission time to permit the sender to receive a CTS reply. Once a sender receives the CTS, it cancels all remaining backoff timers (for other flows) and transmits DS (other motivations for DS have been explained in [12]). When hosts hear either a CTS or a DS message, they will defer until the DATA-ACK transmission completes.

2) Transmission of Packet Based on Sender and Receiver's Table: To schedule a flow, a node should know the flow information of the neighbors of sender and receiver. This information needs to be combined and known by the sender to make the scheduling decision. A straight forward solution would be to broadcast the receiver's table to the sender periodically. However, significant overhead will be induced if the table is large and updated frequently. In our design we assign the backoff time of each flow based on table of both sender and receiver as in [8].

3) Exchanging Slot of an Erroneous Flow with another Flow that Perceives a Good Channel: In the absence of centralized scheduler, it is very difficult to exchange a transmission slot because the node, that is experiencing the errors, cannot convey this message to its two-hop neighbors. We consider a different approach for solving this problem. If the packet of flow f_i, which has the minimum flow tag, is not transmitted within the first minislot, then we consider that the flow is experiencing channel error. So the flow f_j with immediate higher service tag (as mentioned earlier we consider first the lagging flows and then the leading flows in ordering) will transmit the packet. As a result, f_i and f_j will become the lagging and leading flows respectively. Flow f_i will increase its *service_tag* and *lag* value and flow f_j will decrease its *lag* value. But this approach will create another problem, because now the neighbors could not identify whether f_i is experiencing channel errors or it has another neighbor (which is not a neighbor of f_j) with flow that has smaller flow tag than f_i. This problem arises because of the distributed nature of the ad hoc network fair scheduling. Here we generalize both of these two cases. Whatever the reason is as flow f_j is getting chance to transmit before f_i we consider f_i as lagging flow and f_j as leading flow. This assumption will further increase the fairness of the scheduling mechanism.

4) Compensation of Service: A leading flow can releases its extra service, when the flow gets chance to transmit (has the minimum service tag) but its $lag_i < 0$ and it already receives it minimum fraction of service. We assume that all of the two-hop

neighbors have the information regarding flow f_i, it is known to all that f_i should relinquish its extra service in this slot. Therefore, the next flow will be selected from one of the lagging flows (if there is any) and determined based on the *compensation_tag* instead of *service_tag*. Therefore, once a leading flow has the minimum service tag which has to release its extra services, backoff values are assigned only to lagging flows and as a result one of the lagging flows is scheduled to transmit its packet.

5) Propagation of Updated Service Tag: In order to propagate a flow's service tag to all its one-hop neighbors in the node graph and reduce the chance of information loss due to collisions during the propagation, we attach the *service_tag, compensation_tag* and s_i for flow f_i in all four packets RTS, CTS, DS and ACK. However, we do not use the updated tags for flow f_i in RTS and CTS packets, since RTS and CTS do not ensure a successful transmission. When the handshake of RTS and CTS is completed, we attach the updated flow tag in DS and ACK, to inform neighboring nodes about the new updated information of the current transmitting flow f_i.

6 Simulation and Results

In this section, we evaluate our proposed algorithm by simulation and we measure the fairness property of our algorithm. The flow graph used in the simulation is shown in figure 1. There are four flows where each of flow f_0 and f_2 makes contention with all remaining three flows respectively. Reusing of wireless resource allows flow f_1 and f_3 to be transmitted simultaneously and thus increases the throughput of the networks. Each of the flows starts at time 0. Flow f_1, f_2 and f_3 experience error-free wireless link and f_0 experiences a wireless link with error.

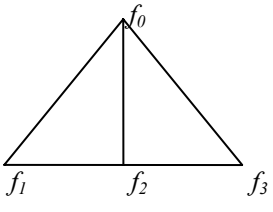


Fig. 1. Example flow graph used in simulation

Figure 2 and Figure 3 show the simulation result of our proposed mechanism. In Figure 2, we consider all the flows are experiencing error-free wireless channel.

In Figure 2(a), all the flows are best-effort flows and have the same weight. So, all flows have got the equal share of the bandwidth. In Figure 2(b), flow f_2 is guaranteed flow and all other flows are best-effort flows. Flow f_2 has the minimum bandwidth requirement of 0.2C, where C is the link bandwidth. So the flow weights are $w_2 = 2$ and $w_0 = w_1 = w_3 = 1$. Accordingly, flow f_2 got double share than the other flows.

In Figure 3, flow f_0 is experiencing an erroneous channel during time interval 4 to 9 seconds and so in this period flow f_0 does not receive any service. Figure 3(a) shows that, at that time f_1 receives the service (lost service of f_0) and as f_1 and f_3 can transmit

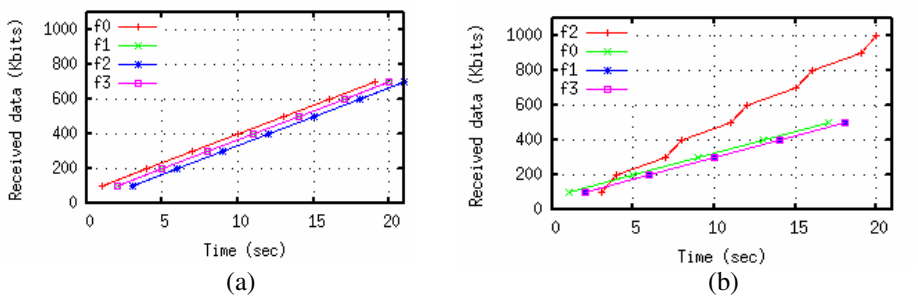


Fig. 2. Simulation results with error-free channel (a) All flows are best-effort (b) Flow f_2 is guaranteed, all others are best-effort

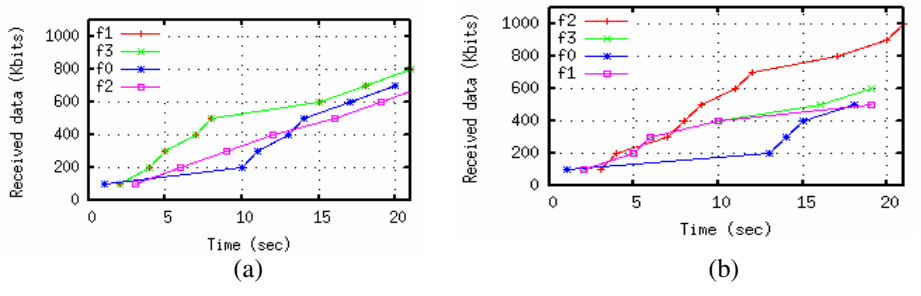


Fig. 3. Simulation results with error in wireless channel (a) all flows are best-effort and (b) flow f_2 is guaranteed flow and all the other flows are best-effort

simultaneously, so f_3 also receives extra service. As the graph shows the lost service of f_0 is compensated at time 11 and 13 seconds, which justify our proposed algorithm.

In Figure 3(b), f_2 is guaranteed flow and all other flows are best-effort flows. And flow f_0 experiences erroneous channel during time interval 4 to 9 seconds and later its service is paid back by flow f_1, f_2 and f_3 those who received the extra service earlier.

7 Conclusions

In this paper, we proposed a distributed fair scheduling algorithm for providing scheduling service in wireless ad hoc networks with errors in wireless link. Our proposed mechanism provides a compensation model for flows that experience an erroneous channel, and allows the flows to get their lost service at a later time. It also ensures that a leading flow can give up its extra service in a graceful way. Also we considered the presence of both QoS and best-effort flows simultaneously in the networks. Our proposed mechanism satisfies the minimum bandwidth requirement of guaranteed flows and provides a fair share of the residual bandwidth to all flows. As a future work, we like to apply our proposed algorithm to mobile ad hoc networks.

References

1. A. Demers, S. Keshav, and S. Shenker: Analysis and simulation of a fair queueing algorithm. ACM SIGCOMM, (1989) 1–12
2. S. Lu, V. Bharghavan, and R. Srikant: Fair scheduling in wireless packet networks. IEEE/ACM Transaction On Networking Vol. 7 (1999) 473–489
3. T. S. Ng, I. Stoica, and H. Zhang: Packet fair queueing algorithms for wireless networks with location-dependent errors. IEEE INFOCOM (1998) 1103–1111
4. P. Ramanathan and P. Agrawal: Adapting packet fair queueing algorithms to wireless networks. ACM MOBICOM (1998) 1–9
5. H. Luo and S. Lu: A self-coordinating approach to distributed fair queueing in ad hoc wireless networks. IEEE INFOCOM (2001) 1370–1379
6. H. Luo and S. Lu: A topology-independent fair queueing model in ad hoc wireless networks. IEEE Int. Conf. Network Protocols (2000) 325–335
7. H. L. Chao and W. Liao: Credit-based fair scheduling in wireless ad hoc networks. IEEE Vehicular Technology Conf. (2002)
8. Jerry Cheng and Songwu Lu: Achieving Delay and Throughput Decoupling in Distributed Fair Queueing Over Ad Hoc Networks. IEEE ICCCN (2003)
9. H. L. Chao and W. Liao: Fair Scheduling With QoS Support in Wireless Ad Hoc Networks. IEEE Trans. on Wireless Comm., vol. 3 (2004)
10. P. Goyal, H.M. Vin and H. Chen: Start-time fair queueing: A scheduling algorithm for integrated service access. ACM SIGCOMM (1996)
11. C. Perkins, E. Belding-Royer and S. Das: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (2003)
12. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang: MACAW: A Medium Access Protocol for Wireless LANs. ACM Ann. Conf. Special Interest Group on Data Comm. (SIGCOMM) (1994)
13. T.S.E. Ng, I. Stoica, H. Zhang: Packet Fair Queueing Algorithms for Wireless Networks with location-Dependent Errors. IEEE INFOCOM (19998)
14. S. Lu, V. Bharghavan and R Srikant: Fair Scheduling in Wireless Packet Networks. IEEE/ACM Transaction. On Networking vol. 7 (1999)
15. H. L. Chao and W. Liao: Fair Scheduling in Mobile Ad Hoc Networks with Channel Errors. IEEE Transaction of Wireless Communications, Vol 4 (2005)
16. M. M. Alam, M. M. Rashid and C. S. Hong: Distributed Coordination and Fair Queueing in Wireless Ad Hoc Networks. Lecture Notes in Computer Science, Vol. 3981 (2006)

Performance Analysis of Service Differentiation for IEEE 802.15.4 Slotted CSMA/CA*

Meejoung Kim

Research Institute for Information and Communication Technology, Korea University
1, 5-ga, Anam-dong, Sungbuk-gu, 136-701, Seoul, Korea
meejkim@korea.ac.kr

Abstract. In this paper, we propose two mechanisms, differentiation by backoff exponent and differentiation by the size of contention window, for IEEE 802.15.4 sensor networks to provide multi-level differentiated services in beacon-enabled mode with slotted CSMA/CA algorithm under non-saturation condition. Mathematical model based on discrete-time Markov chain is presented and analyzed to measure the performance of the proposed mechanisms. Numerical results show that the delicate tuning of throughput could be performed by differentiation by backoff exponent while differentiation by the size of contention window gives more effect for differentiation of throughput.

Keywords: sensor networks, service differentiation, Markov chain.

1 Introduction

The success of wireless sensor networks as a technology depends on the success of the standardization efforts to unify the market and avoiding the proliferation of proprietary and incompatible protocols that will limit the size of overall wireless sensor market. Since IEEE 802.15.4 standard for low data rate wireless personal area networks (LR-WPAN) supports small, inexpensive, energy efficient devices operating on battery power that require no infrastructure to operate, it is considered as one of the technology candidates for wireless sensor networks [1]-[2].

The standard supports two network topologies, the peer-to-peer topology in which devices can communicate with one another directly as long as they are within the physical range and the star topology in which devices must communicate through a central controller device commonly referred as PAN coordinator. It also defines two channel access mechanisms depending on whether a beacon frame is used to synchronize communications or not. In the beacon enabled networks, slotted carrier sense multiple access mechanism with collision avoidance (CSMA/CA) is used and the slots are aligned with the beacon frame which is sent periodically by the PAN coordinator. On the other hand, in the non-beacon enabled networks, unslotted

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 11 – 22, 2006.
© Springer-Verlag Berlin Heidelberg 2006

CSMA/CA is used and no beacon frame is used. Since the central device manages the networks in general and the PAN coordinator can act as the network controller and the sink to collect data from sensor nodes, for sensor network implementation, the star topology operating in beacon enabled mode appears to be better suited than the peer-to-peer topology operating in non-beacon enabled mode.

The related researches deal with the operation of PAN operating under 802.15.4 in the beacon enabled mode under saturation condition [2-6] and non-saturation [7].

In sensor network, multiple queues cannot be used and only packets with small size are transmitted dues to the characteristics of the devices. Therefore, the priorities have to be considered in a slightly different way compare to those using multiple queues in one node. In this paper, we propose two mechanisms for modified 802.15.4 sensor networks which provide multi-level differentiated services for each and every device. To investigate the performance of the proposed mechanisms, mathematical model which is based on discrete-time Markov chain is presented. By analyzing the Markov chain, we obtain the probabilities that the medium is idle when the device is sensing, the probabilities of attempting transmission, throughputs, delays, and drop probabilities for different priority classes.

The rest of the paper is organized as follows. In Section 2, we propose the operating mechanisms, and the mathematical model for the proposed mechanisms is presented in Section 3. Even though several performance measures such as delay and packet drop probability can be considered, we present the throughput performance only in Section 4. Section 5 provides the numerical results.

2 Priority-Based Service Differentiation Scheme

In this section, we propose two mechanisms for service differentiation. The 802.15.4 sensor network model that we are considering is operated in the beacon-enable mode with slotted CSMA/CA algorithm. The structure of the superframe under 802.15.4 is illustrated in Fig. 1 and we consider contention access period (CAP) portion only under the non-saturation mode. We assume there are Q+1 different priorities classes and a wireless PAN with n nodes which is composed of n_q nodes

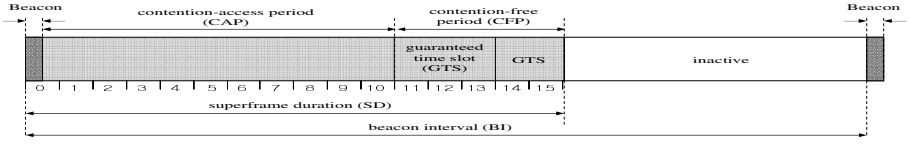


Fig. 1. The structure of the superframe under IEEE 802.15.4

in each priority q class, in other words $n = \sum_{q=0}^{Q} n_q$, where q denotes the priority taking integer values in $[0,Q]$. Throughout the paper, we assume that each and every device has its own priority which does not change during a superframe and the packet transmission is completed during the superframe. We assume that packets are

generated by Poisson process with arrival rate λ_q for each device in priority q class and each device has a finite buffer so that the newly generated packets are dropped when the buffer is fully occupied. Then the probability that the device does not have an empty queue after transmit a packet is given by $\min(1, \lambda_q E(D_q))$, where $E(D_q)$ is the average service time of priority q class. Denote it by ρ_q and note that $\rho_q = 1$ implies the saturation mode. Furthermore, we assume that if the transmitted packet collides, the packet will be dropped and the device will try to transmit a new packet in the head of the queue if there is a packet waiting for transmission.

Since we conjectured that the parameters, the size of contention window and backoff exponent, will give different effects on IEEE 802.15.4 wireless sensor network performances, we consider two service differentiation mechanisms by varying values of these two parameters. In both differentiation mechanisms, we apply the scheme which chooses the random backoff counter differentially in each stage. In other words, during the ith backooff stage, the backoff counter is randomly chosen in $[W_{i-1}, W_i\text{-}1]$ rather than $[0, W_i\text{-}1]$, $W_i=2^i W_0$.. This scheme may reduce the collisions by waiting more time before sensing and gives better chance to access the channel to a device with early backoff stage. Even though the combined mechanism of the proposed two mechanisms can be considered, we consider the mechanisms separately in this paper. In the following subsection, we describe the mechanisms in detail.

2.1 Service Differentiation by the Size of Contention Window (CW)

Let the priority of classes are given by

$$\text{classQ} \prec \text{class(Q-1)} \prec \cdots \prec \text{class0}, \tag{1}$$

where $\prec$ is the symbol for the order of priority, in other words class0 and classQ are the highest and the lowest priority classes, respectively. Then we differentiate the CW value of each service class, class(q), by CW[q] which satisfies the following relation:

$$CW[0] < CW[1] < \cdots < CW[Q] \tag{2}$$

The relation in Eq. (2) is intuitively clear since a device with smaller CW value will take better chances of transmission than the device with larger CW value in general.

Compare to conventional 802.15.4 MAC protocol where every device initiates the CCA procedure with the same CW value, the proposed mechanism performs the CCA procedure with different CW value according to the request of the device.

2.2 Service Differentiation by Backoff Exponent (BE)

Differentiation by BE is as similar as differentiation by CW. In this mechanism, we differentiate the BE value of each service class, class(q), by BE[q]. With the relation of Eq. (1), the corresponding BE values have to satisfy the following relation:

$$BE[0] < BE[1] < \cdots < BE[Q] \tag{3}$$

The relation in Eq. (3) is also intuitively clear since devices with smaller BE values will take better chances to access the channel than those with larger BE values. The difference between conventional 802.15.4 and the proposed mechanism begins at the backoff stage 0 by choosing a random backoff counter in different range.

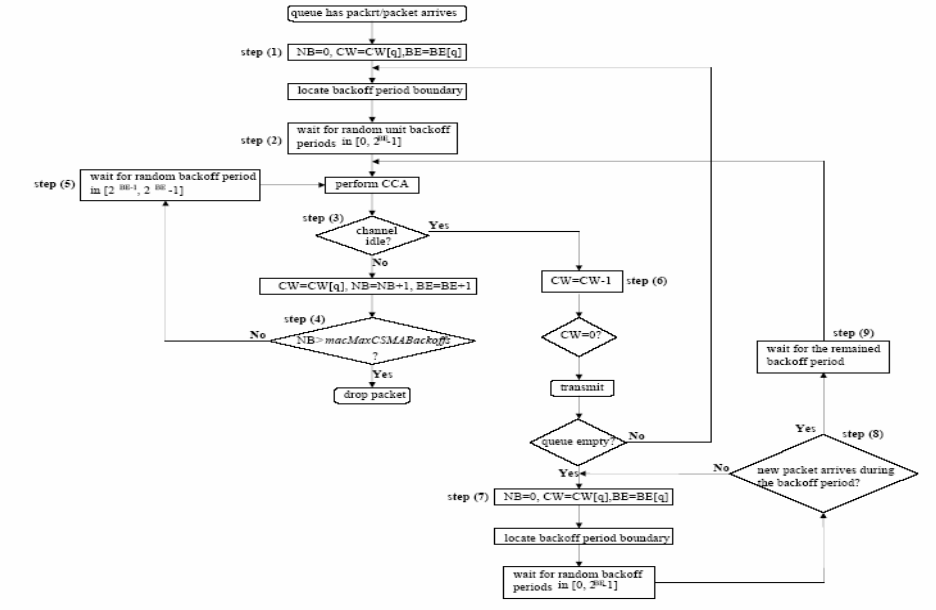


Fig. 2. Operation of the priority-based service differentiation mechanisms

2.3 Operation of Priority-Based Service Differentiation Scheme

Fig. 2 describes the operation of the proposed scheme. Due to the space limitation, we do not explain the operation step by step.

3 Analytical Model

To analyze the proposed mechanisms, we introduce the following two stochastic processes for a given device in the priority q class. Let $n(q, t)$ and $b(q, t)$ be the stochastic processes representing the values of NB and the value of the backoff counter and/or contention window, respectively, at time t and denote e for empty buffer. Note that NB represents the backoff stage within the range of $[0, m+1]$, m = *macMaxCSMABackoffs*. Then $\{(n(q, t), b(q, t), e), (n(q, t), b(q, t))\}$ forms a multi-dimensional Markov process defining the state of the head packet in the queue at the backoff unit boundaries. Since we are assuming that every and each device has its own priority which does not change in a superframe and the packet transmission is completed in the superframe, without loss of generality, each of the processes $n(q, t), b(q, t)$ can be written simply as $n(t), b(t)$ during the superframe. Then the corresponding state space is denoted as follows:

$$\Omega = \{(0, b_0(t), e), (n(t), b(t)), (m+1, 0) \mid 0 \le n(t) \le m,\ 0 \le b_0(t) \le W_0 - 1,$$
$$-CW[q] \le b(t) \le W_i - 1,\ i = 0, \cdots, m,\ n(t),\ b_0(t), \text{ and } b(t) \text{ are integers}\},$$

where $W_0 = 2^{BE[q]}$, $W_i = 2^i W_0$, and $(0, b_0(t), e)$ is the state of the device with empty buffer. In the proposed model, when the queue is empty the backoff counter is chosen randomly in [0, W_0-1]. If a device is in the state *(0,i,e)* and the new packet arrives, the state is changed to *(0,i)*. If the new packet does not generated until the state becomes *(0,0,e)*, the backoff counter is choose again randomly in [0, W_0-1]. In addition, the state (i, j), $i \in [0, m]$, $j \in [0, W_i - 1]$, and $(i, -j)$, $i \in [0, m]$, $j \in [1, CW[q]]$, denote the device in the mode of decreasing backoff counter and sensing the channel, respectively.

Let δ and $N(\delta)$ be the unit of backoff period and the number of packets arrived during the backoff period, respectively. Then the probability that at least one packet arrives in δ time interval is given by

$$P_q \equiv 1 - e^{-\lambda_q \delta}. \tag{4}$$

The state transition diagram of these states is illustrated in Fig 3. For the simplicity of the notations, we use the transition probabilities $P(i_1, j_1 \mid i_0, j_0)$ instead of

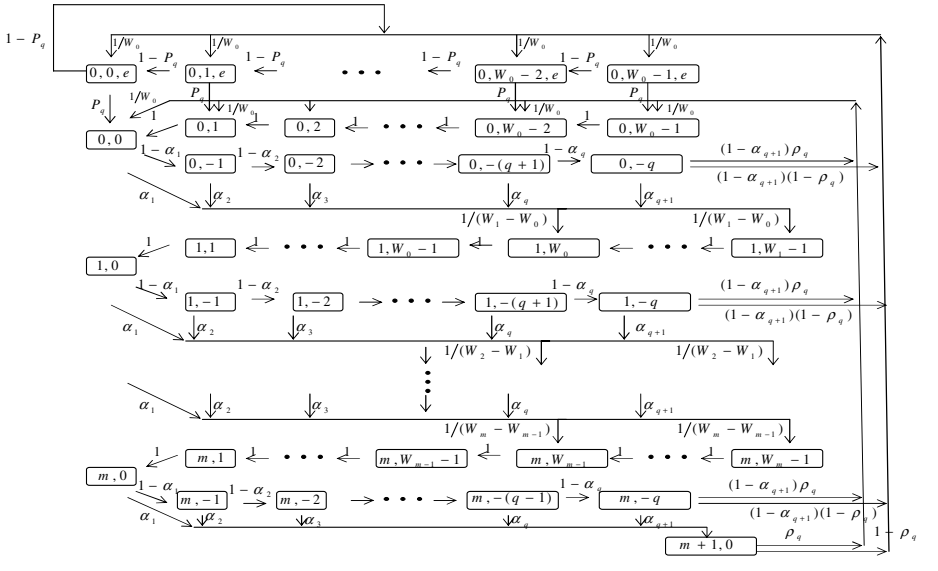


Fig. 3. Markov chain model

$P(n(t+\delta) = i_1, b(t+\delta) = j_1 \mid n(t) = i_0, b(t) = j_0)$. Then the one-step transition probabilities are given as follows:

$$P(0, j-1, e \mid 0, j, e) = 1 - P_q, \quad P(0, j, e \mid 0, 0, e) = (1 - P_q)/W_0, \; j \in [0, W_0 - 1], \tag{5}$$

$$P(0, j \mid 0, j, e) = P_q, \; P(0, j \mid m+1, 0) = \rho_q / W_0, \; j \in [0, W_0 - 1], \tag{6}$$

$$P(i, j-1 \mid i, j) = 1, \; i \in [0, m], j \in [1, W_i - 1], \tag{7}$$

$$P(i, -(j+1) \mid i, -j) = 1 - \alpha_{j+1}, \; i \in [0, m], j \in [0, CW[q] - 1], \tag{8}$$

$$P(i+1,W_{i+1}-k \mid i,-j)=\alpha_{j+1}/(W_{i+1}-W_i),$$
$$i\in[0,m-1], j\in[1,CW[q]],\ k\in[1,W_{i+1}-W], \quad (9)$$

$$P(0,j \mid i,-CW[q])=(1-\alpha_{q+1})\rho_q/W_0,\ i\in[0,m], j\in[0,W_0-1], \quad (10)$$

$$P(0,j,e \mid i,-CW[q])=(1-\alpha_{q+1})(1-\rho_q)/W_0,\ i\in[0,m], j\in[0,W_0-1], \quad (11)$$

$$P(0,j,e \mid m+1,0)=(1-\rho_q)/W_0,\ j\in[0,W_0-1], \quad (12)$$

and $P(m+1,0 \mid m,-j)=\alpha_{j+1},\ j\in[0,CW[q]]$. (13)

In the equations, $\alpha_j, j=1,\cdots,CW[q]+1$, is the probability of the channel be busy in jth CCA procedure at the states $(i,-j),\ i=1,\cdots,m,\ j=0,\cdots,CW[q]$. To find out these probabilities, we consider the probabilities $P_{I,n-1}$, $P_{S,n-1}$, and $P_{C,n-1}$ which are the probabilities of channel idle, successful transmission of other device, and collision of other devices, respectively, when a device in q class is sensing. Then we have the following equations:

$$P_{I,n-1}=\prod_{i=0,i\neq q}^{Q}(1-\tau_i)^{n_i}(1-\tau_q)^{n_q-1}=\frac{1}{1-\tau_q}\prod_{i=0}^{Q}(1-\tau_i)^{n_i}, \quad (14)$$

$$P_{S,n-1}=\sum_{i=0}^{Q}n_i\tau_i(1-\tau_i)^{n_i-1}\prod_{j=0,j\neq i}^{Q}(1-\tau_j)^{n_j}(1-\tau_q)^{-1}-\tau_q(1-\tau_q)^{n_q-2}\prod_{i=0,i\neq q}^{Q}(1-\tau_i)^{n_i} \quad (15)$$

and $P_{C,n-1}=1-P_{I,n-1}-P_{S,n-1}$, (16)

where τ_q is the probability that a device in the priority q class transmits during a backoff period and ρ_q is the probability of the queue is not empty which is given in Section 2.

Let T_S and T_C be the average durations for successful transmission and collision after the final CCA procedure, respectively. Then T_S and T_C are given by

$$T_S=T_H+T_{packet}+t^s_{wack}+t_{ack} \text{ and } T_C=T_H+T_{packet}+t^f_{wack}, \quad (17)$$

where T_H, T_{packet}, t^s_{wack}, t_{ack}, and t^f_{wack} are the average durations of transmitting the header (including MAC header and PHY header), packet transmission, waiting time for ACK for successful transmission, time for receiving ACK, and waiting time for ACK for unsuccessful transmission. Then the probability of the channel busy in the 1st CCA procedure which is determined by either successful transmission of one device or collision out of n-1 devices is given by

$$\alpha_1=\frac{P_{S,n-1}T_S+P_{C,n-1}T_C}{P_{I,n-1}\delta+P_{S,n-1}T_S+P_{C,n-1}T_C} \quad (18)$$

and subsequently we obtain the following probabilities of channel idle during the jth CCA procedure:

$$1-\alpha_j=P_{I,n-1}{}^{j-1}(1-\alpha_1),\ j=1,\cdots,CW[q]+1 \quad (19)$$

Since all of the states in Ω is positive recurrence and the system is stable, there exist the stationary probabilities $\{b_{0,j_0,e}, b_{i,j}, b_{m+1,0} : i \in [0,m], j_0 \in [0,W_0-1], j \in [-CW[q], W_i-1]\}$ of the discrete-time Markov chain which is defined by

$$b_{0,j,e} = \lim_{t\to\infty} P(0, b(t) = j, e), \; b_{i,j} = \lim_{t\to\infty} P(n(t) = i, b(t) = j),$$

and $$b_{m+1,0} = \lim_{t\to\infty} P(n(t) = m+1).$$

Let $\mathbf{b}$ be the stationary vector, i.e.

$\mathbf{b} = (b_{0,0,e}, b_{0,1,e}, \cdots, b_{0,W_0-1,e}, b_{0,-q}, b_{0,-(q-1)}, \cdots, b_{0,0}, b_{0,1}, \cdots, b_{0,W_0-1}, b_{1,-q}, \cdots, b_{1,W_1-1}, \cdots, b_{m,-q}, \cdots, b_{m,W_m-1}, b_{m+1,0})$.

Then it satisfies the following equations:

$$\mathbf{bP = b} \text{ and } \sum_{i=0}^{m}\sum_{j=-q}^{W_i-1} b_{i,j} + \sum_{j=0}^{W_i-1} b_{0,j,e} + b_{m=1.0} = 1, \tag{20}$$

where $\mathbf{P}$ is the transition probability matrix when Ω is ordered lexicographically. To simplify the equations, let $A_k \equiv \sum_{l=1}^{k} \alpha_l \left(\prod_{r=1}^{l-1} (1-\alpha_r) \right)$. Then by Eq. (20) and through complicated calculation, we obtain the following relations:

$$b_{i,0} = b_{i,1} = \cdots = b_{i,W_{i-1}} = A^i_{CW[q]+1} b_{0,0}, i = 1, \cdots, m, \tag{21}$$

$$b_{m+1,0} = \sum_{l=1}^{CW[q]+1} \alpha_l b_{m,-(l-1)} = A^{m+1}_{CW[q]+1} b_{0,0} \tag{22}$$

$$b_{i,j} = \frac{W_i - j}{W_i - W_{i-1}} A^i_{CW[q]+1} b_{0,0}, i = 1, \cdots, m, \; j = W_{i-1}, \cdots, W_i - 1, \tag{23}$$

$$b_{i,-j} = \prod_{r=1}^{j} (1-\alpha_r) A^i_{CW[q]+1} b_{0,0}, i = 1, \cdots, m, \; j = 1, \cdots, CW[q], \tag{24}$$

$$b_{0,j,e} = \frac{1-(1-P_q)^{W_0-j}}{P_q} (B b_{0,0} + \frac{1}{W_0}(1-P_q) b_{0,0,e}), \; j = 0, \cdots, W_0 - 1, \tag{25}$$

where B is given by $B = \frac{1}{W_0}(1-\rho_q) \prod_{r=1}^{CW[q]+1} (1-\alpha_r) \sum_{i=0}^{m} A^i_{CW[q]+1} + \frac{1}{W_0}(1-\rho_q) A^m_{CW[q]+1}$. Furthermore, $b_{0,0}$ is calculated as

$$b_{0,0} = \left(\frac{1-(1-P_q)^{W_0}}{P_q} B \right)^{-1} \left(1 - \frac{1-(1-P_q)^{W_0}}{P_q} \frac{1}{W_0}(1-P_q) \right) b_{0,0,e}. \tag{26}$$

Denote $\left(\frac{1-(1-P_q)^{W_0}}{P_q} B \right)^{-1} \left(1 - \frac{1-(1-P_q)^{W_0}}{P_q} \frac{1}{W_0}(1-P_q) \right)$ in Eq. (26) by Ξ_1 for notational simplicity. Then substituting Eq. (26) into Eq. (21)-Eq. (25), $b_{0,j,e}$,

$j \in [1, W_0 - 1]$, $b_{i,j}$, $i \in [1, m]$, $j \in [-CW[q], W_i - 1]$, $b_{m+1,0}$ can be expressed by $b_{0,0,e}$. Therefore, substituting these into the 2nd part of Eq. (20), we obtain $b_{0,0,e}$ as follows:

$$b_{0,0,e} = \left[\left(B\Xi_1 + \frac{1}{W_0}(1-P_q) \right) \cdot \left(\frac{1}{P_q}(1 + P_q W_0 - P_q) + \frac{(1-P_q)^2 - (1-P_q)}{P_q^2} + \frac{1}{2} W_0 (W_0 - 1)(1 + C_1 \Xi_1) \right) \right.$$
$$\left. \cdot \left\{ \left(1 + \frac{A_{q+1} - A_{q+1}^{m+2}}{1 - A_{q+1}} \right) + e(m) + d(m) + C(q) \cdot \left(1 + \frac{A_{q+1} - A_{q+1}^{m+1}}{1 - A_{q+1}} \right) \right\} \cdot \Xi_1 \right]^{-1}, \quad (27)$$

where $C(q), C_1, d(m),$ and $e(m)$ are defined by $C(q) = \sum_{j=1}^{q} \prod_{r=1}^{j} (1 - \alpha_r)$, $C_1 = \frac{1}{W_0} \rho_q \prod_{r=1}^{q+1} (1 - \alpha_r) \sum_{i=0}^{m} A_{q+1}^i + \frac{1}{W_0} \rho_q A_{q+1}^{m+1}$, $d(m) = \sum_{i=1}^{m} \sum_{j=W_{i-1}}^{W_i - 1} \frac{W_i - j}{W_i - W_{i-1}} A_{q+1}^i$, and $e(m) = \sum_{i=1}^{m} W_{i-1} A_{q+1}^i$, respectively. By substituting Eq. (27) into Eq. (26), we obtain $b_{0,0}$ and subsequently the stationary probabilities $\{b_{0,j_0,e}, b_{i,j}, b_{m+1,0} : i \in [0, m], j_0 \in [0, W_0 - 1] j \in [-CW[q], W_i - 1]\}$ from Eq. (21)-Eq. (25). With these stationary probabilities, we find the probability that a device in the priority q class transmits during a unit backoff period is given by $\tau_q = \sum_{i=0}^{m} b_{i,-q}$.

4 Performance Analysis

4.1 Throughput

Let p_S and $p_{S,q}$ be the probabilities that a successful transmission occurs by any priority class and a successful transmission occurs by a device in the priority q class in a unit backoff period, respectively. Then these probabilities are calculated as follows:

$$p_S = n \sum_{k=0}^{Q} \frac{\tau_k}{1 - \tau_k} \prod_{h=0, l \neq k}^{Q} (1 - \tau_h)^{n_h} \text{ and } p_{S,q} = \frac{n_q \tau_q}{1 - \tau_q} p_I, \; q \in [0, Q],$$

where p_I is the probability of channel idle which is given by

$$p_I = \prod_{k=0}^{Q} (1 - \tau_k)^{n_k}, \; n = \sum_{k=0}^{Q} n_k. \quad (28)$$

Let p_B be the probability that the channel is sensed busy in a unit backoff period which is given by $p_B = 1 - p_I$. Then $p_B - p_S$ is the probability that the channel is sensed busy by collisions that occur from any priority class. In addition, the

probability that a collision occurs in a unit backoff period for the priority q class which is denoted by $p_{C,q}$ is given by

$$p_{C,q} = 1-(1-\tau_q)^{n_q-1} \prod_{k=0,k\neq q}^{Q} (1-\tau_k)^{n_k},\ q \in [0,Q].$$

Let S_q be the normalized throughput for the priority q class. Then we can express the normalized throughput S_q as the following ratio:

$$S_q = \frac{p_{S,q}T_{packet}}{P_I\delta + p_S T_S + (p_B - p_S)T_C}$$

4.2 Average Delay and Packet Drop Probability

Average delay and packet drop probability are other measures for the performance. In this paper, however, we do not present them owing to the space limitation.

5 Numerical Results

In this section we present the performance of the analytical results, which show the effect of the variations of CW[q] and BE[q] for service differentiation. We considered BPSK mode and used the parameters in 802.15.4 standard, which are listed in Table 1.

For the numerical result, we use all of the assumptions described in Sec. 2-3 such as Poisson process of packet arrival and Q is taken by 2, in other words, 3 different priority classes are considered. The arrival rate of each class is set by the same value 900 symbols during the time of unit backoff period which is 20/bpsk rate. Such an arrival rate gives the probability of a new packet arrival by 0.5934.

For the service differentiation by BE, the value of CW[q] is set by 2 for all the devices in every class and the values of BE[0], BE[1], and BE[2] at the device of each class are set by 1, 2, and 3, respectively. On the other hand, for the service differentiation by CW, we set BE[q] is set by 1 for all devices and the values of CW[0], CW[1], and CW[2] are set by 2,4,8, respectively. In the following figures, x axis denotes the number of devices of each class and we assume that the number of devices in each class is all the same.

Table 1. Parameter set used in the numerical analysis

packet payload	127x8 bits	channel bit rate	20 Kbit/s
time for waiting and receiving ACK for successful transmission	32 symbols	MAC header	200bits
time for aiting ACK for collision	54 symbols	PHY header	48 bits
retry limit	5	unit backoff period	20 symbols

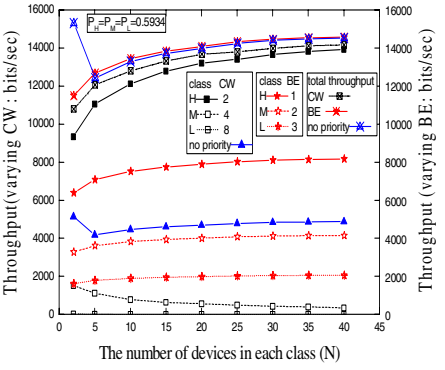


Fig. 4. Comparison of throughputs

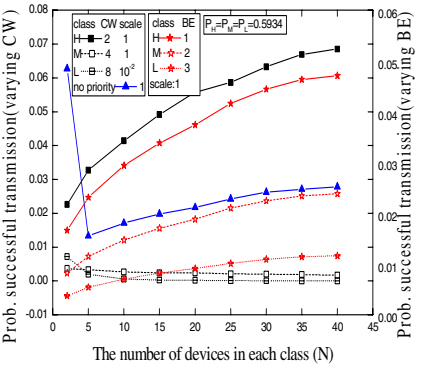


Fig. 5. Comparison of prob. of successful trans

The non-saturation throughputs for these values of CW[q] and BE[q] are shown in Fig. 4. In addition, the throughput of no differentiation with CW=2 and BE=1 for all devices and total throughputs of all of these are added. As see in the figure, for BE differentiation, the throughput of a device on each class slightly increases as the total number of devices increases and that of high class outperforms that of no priority considered. On the other hand, for CW differentiation, it decreases for middles and lower classes while that of high class increases regardless of the number of the devices. It implies that the devices of lower classes yield the opportunity to occupy the channel to the devices in higher class for CW differentiation. The delicate tuning of the throughput could be performed by varying the value of BE[q] while the value of CW[q] could be adjusted to increase the throughput of the high class. The total throughputs of these cases have the following relation, except $N = 2$, even though the differences of performances are negligible: $\text{total}S_{CW} < \text{total}S_{\text{no piority}} < \text{totalS}_{BE}$. For $N = 2$, we note that the total throughput of the case of no priority considered is 15,408 bits/sec while those of BE and CW differentiations are 11,596 bits/sec and 10,891 bits/sec, respectively. It dues to the fact that τ_q and the success probabilities out of τ_q have the following relations:

$$\tau_{H,BE}(0.0025) < \tau_{H,CW}(0.0115) < \tau_{\text{no priority}}(0.038) \tag{29}$$

$$p_{S,H,CW}(0.0226) < p_{S,H,BE}(0.0347) < p_{S,\text{no priority}}(0.0627) \tag{30}$$

where $\tau_{H,\cdot}$ and $p_{S,H,\cdot}$ are the transmission probability and the probability of successful transmission out of the transmission probability of high class in each algorithm, respectively. In these relations, we note that for the higher class the more chances of transmissions are given in CW differentiation, but the more successful transmissions occur in BW differentiation.

Fig. 5 shows the probability of successful transmissions of three different service classes. It shows that success probabilities increase as the number of devices increase regardless of the priority classes for BE differentiation. On the other hand, for CW differentiation, that of the high class is increasing while those of middle and low

classes are decreasing as the number of devices is increasing. It implies the service differentiation is greatly effected by CW differentiation rather than BE differentiation just as same as throughput.. It is note worthy that the both probabilities of trying to transmit and subsequent successful transmission with no priority are the highest values compare to those values with differentiations are considered when $N = 2$. This is due by Eq. (29)-Eq. (30) and the fact that the probability of idle channel without priority is lower than those with priority even though we do not present in this paper. Actually, $p_{I,\text{no priority}} = 0.7924$ while $p_{I,CW} = 0.9734$ and $p_{I,BE} = 0.9684$, where $p_{I,\bullet}$ is given by Eq. (28) when $\bullet$ algorithm is applied. Regardless of the number of devices, the relations $\tau_{H,CW} > \tau_{H,BE}$, $\tau_{M,CW} < \tau_{M,BE}$, and $\tau_{L,CW} > \tau_{L,BE}$ hold, where H, M, and L are the high, middle, and low priority classes, respectively.

6 Conclusion

In this paper, we proposed two mechanisms for IEEE 802.15.4 sensor networks which provide multiple level differentiated services for each and every device. The mathematical model based on discrete-time Markov chain is provided for analyzing the performance of the proposed mechanisms.

Numerical results show that the variation of contention window size has more effect on service differentiation than that of backoff exponent as the number of devices increase. The delicate tuning of the throughput could be performed by varying backoff exponent while the better throughput of the high class could be performed by adjusting contention window size.

For the future work, we will find out the optimal numbers of devices for different packet arrival rates in the viewpoint of maximizing throughput and of minimizing delay, which will provide a criterion for using the parameters for specific purposes.

References

[1] Standard for part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless personal Area Networks (WPAN), IEEE std. 802.15.4, IEEE, NY, 2003.

[2] J. Misic, S. Shafi, and V. B. Misic, The Impact of MAC Parameters on the Performance of 802.15.4 PAN, Elsevier Ad hoc Networks, vol. 2, pp. 351-371, July 2004.

[3] J. Misic, S. Shafi and V. B. Misic, Performance of a Beacon Enabled IEEE 802..15.4 Cluster with Downlink and Uplink Traffic, IEEE Transaction on Paralle; and Distributed Systems, vol. 17, no. 4, pp 361-376, April 2006.

[4] T. R. Park, T. H. Kim, J. Y. Choi, S. Choi, and W. H. Kwon, Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA, Eelectronics Letters, vol. 41, no. 18, 1st Sep. 2005.

[5] T. H. Kim and S. Choi, Priority-based Delay Mitigation for Event-Monitoring IEEE 802.15.4 LR-WPANs, IEEE Communication Letters, vol. 10, issue 3, pp. 213-215, Mar. 2006.

[6] E.-J. Kim, M. Kim, S.-K. Youm, S. Choi, and C.-H. Kang, Priority-Based Service Differentiation Scheme for IEEE 802.15.4 Sensor Networks, will be published in AEU-International Journal of Electronics and Communications, 2006.
[7] J. Misic and V. B. Misic, Access delay for nodes with finite buffers in IEEE 802.15.4 beacon enabled PAN with uplink transmissions, Computer Communications, vol. 28, pp1152-pp1166, 2005.
[8] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehanene, Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modelling and Improvement Perspective, Design, Automation and Test in Europe(DATE'05), vol. 1, pp.196-201, 2005.

Information-Driven Task Routing for Network Management in Wireless Sensor Networks*

Yu Liu[1], Yumei Wang[1], Lin Zhang[1], and Chan-hyun Youn[2]

[1] School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing, China
liuyurainy@gmail.com, {ymwang, zhanglin}@bupt.edu.cn
[2] Information and Communications University, Republic of Korea
chyoun@icu.ac.kr

Abstract. Wireless sensor networks (WSNs) consist of a large collection of small nodes providing collaborative and distributed sensing ability in unpredictable environments. Given their unattended nature, it is important for the routing algorithm in WSN to deal well with the node failures, resource depletion, and other abnormality. In this paper, an information-driven task routing (IDTR) is introduced for WSN with their specific application to network management. In WSN, a query for the location of the target may be launched from a proxy, which requires the network to collect and transfer the information to an exit node, while keeping the whole communication cost small. The proposed IDTR-based heuristic search algorithm evaluates the expected mutual information of the neighbor nodes, and selects the one that has the maximum information contribution and satisfies the constraint of communication cost as the next hop. In order to avoid the effects of any possible sensor holes, it will calculate the information contribution of all the M hop neighbor nodes to bypass the holes. Simulation results show that IDTR-based heuristic search algorithm achieves the tradeoff between the data communication cost and the information aggregation gain.

Keywords: Information-driven task routing, information contribution, sequential Bayesian filtering, heuristic search algorithm.

1 Introduction

Wireless sensors have the ability of sensing, computation and communication, and work under the ad hoc mode to compose wireless sensor networks. Due to the small volume of sensors, there are many resource constraints such as limited on-board battery power and limited communication bandwidth, etc.. Because of the characteristics in sensing ability and spatial coverage, WSN is ideally suited for tracking moving targets, monitoring a large number of objects, detecting low-observable events, and estimating the status of a target. The routing of these tasks is

* Supported by National Science Foundation of China (NSFC) with contract number (60502036).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 23 – 32, 2006.
© Springer-Verlag Berlin Heidelberg 2006

not only to transfer data from one point to another, but should also be optimized for the data transmission and the information aggregation.

This paper introduce an information-driven task routing algorithm to solve the special routing scenario with communication cost constraints and sensor holes. It utilizes the concept of information utility function to evaluate the expected mutual information of the neighbor nodes, and selects the one that has the maximum information contribution and satisfies the constraint of communication cost as the next hop. And it calculates the information contribution of all the M hop neighbor nodes to bypass the sensor holes.

Many algorithms were proposed to address the routing problem in WSNs. The energy-aware routing [1] selected the path to minimize the energy exhaustion chance of the sensors. Directed diffusion [2] routed data based on low-level data attributes rather than sensor addresses. For the holes routing problem, the memorization based algorithms [3] required nodes to record their past traffic or paths, and GPSR [4] followed the perimeter of the hole to bypass it.

The idea of utilizing the information utility to manage sensing resources has been investigated in computer fields already. [5] introduced an expected utility measure for decentralized sensing system based on local decision. [6] proposed the information-directed routing algorithm to track targets in WSNs, which illumines us to explore the applications of the information-driven task routing (IDTR) algorithm for the routing problems with communication cost constraints and sensor holes.

This paper is organized as follows. In section 2, the routing scenario of tracking the target, the sequential Bayesian filtering, the information utility function and the two related routing algorithms are introduced, and the IDTR-based heuristic search algorithm is proposed. Section 3 is the simulation results and analyses.

2 Information-Driven Task Routing

2.1 Routing Scenario

For the application of target tracking in WSNs, the user may initiate a query from a query proxy, requiring the sensor network to collect the target information and estimate its location. The query proxy has to figure out where such information can be collected and routes the query towards the high information content region, as illustrated by the co-centric ellipses in Fig.1. This differs from the routing in traditional communication networks where the destination is often known to the proxy. Here, the destination is unknown and is dynamically determined. So the geographical routing protocols are not suited.

Sometimes the aggregated information is required to be transferred to an exit node, where it can be extracted for further processing, and the total communication cost is also limited to be no more than a specified constraint [7], as illustrated in Fig.2. Furthermore, the sparse distribution of sensors or the energy exhaustion of some sensors may result in sensor holes. As illustrated in Fig.3, the two sensors in the middle run out their deployed energy, resulting in a sensor hole as the dotted ellipse area.

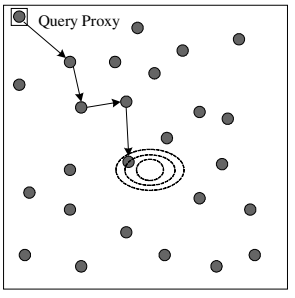


Fig. 1. Query to the high information region

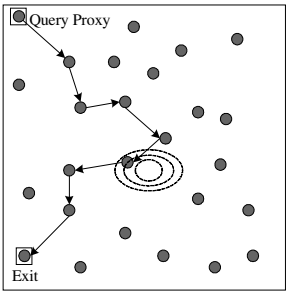


Fig. 2. Query To Whom It May Concern: the exit node

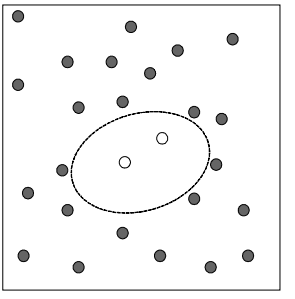


Fig. 3. Sensor hole exists

2.2 The Foundation of Information-Driven Task Routing

For the tracking task in WSNs, we want to estimate the current position of the target $x^{(t)}$ based on the measurements of different sensors at different steps $\left\{z_{i_0}{}^{(0)}, z_{i_1}{}^{(1)}, \cdots, z_{i_t}{}^{(t)}\right\}$. IDTR utilizes sequential Bayesian filtering to estimate the location of the target, evaluates the expected mutual information of the neighbor nodes, and selects the one with maximum information contribution as the next hop.

2.2.1 Sequential Bayesian Filtering

Sequential Bayesian filtering [8] is a generalization of the well-known Kalman filtering [9]. At time t, some rough *a priori* knowledge (called belief) about where the target is known, usually in the form of a probability mass function $p\left(x^{(t)} / z^{(1)}, \cdots, z^{(t)}\right)$. At time $t+1$, the sequential Bayesian filtering incorporates the measurement $z_j^{(t+1)}$ to update the belief as follows:

$$p\left(x^{(t+1)} / \overline{z}^{(t+1)}\right) = C \cdot p\left(z^{(t+1)} / x^{(t+1)}\right) \cdot \int p\left(x^{(t+1)} / x^{(t)}\right) \cdot p\left(x^{(t)} / \overline{z}^{(t)}\right) dx^{(t)} \tag{1}$$

where $p\left(x^{(t)} \mid \overline{z}^{(t)}\right)$ is the prior belief given a history of the measurements up to time t, $\overline{z}^{(t)} = \left\{z^{(0)}, \cdots, z^{(t)}\right\}$, $p\left(x^{(t+1)} \mid x^{(t)}\right)$ specifies the predefined dynamics model of the target, $p\left(z_j{}^{(t+1)} \mid x^{(t+1)}\right)$ is the likelihood function of the measurement of the sensor j, and C is a normalization constant. Typically, the expectation value of the distribution $\overline{x}^{(t)} = \int x^{(t)} p\left(x^{(t)} \mid \overline{z}^{(t)}\right) dx^{(t)}$ is considered as the estimated location of the target, and the variance $\Sigma = \int \left(x^{(t)} - \overline{x}^{(t)}\right)\left(x^{(t)} - \overline{x}^{(t)}\right)^T p\left(x^{(t)} \mid \overline{z}^{(t)}\right) dx^{(t)}$ is the residual uncertainty.

2.2.2 Information Utility Function

During the path selection, the sensors on the path can contribute to successive message refinement in tracking applications. We may select the next hop based on the rules of shortest distance, or the rule of maximum remained energy. Here we mainly discuss the principle of maximum information contribution [10].

We adopt the mutual information [11] to quantify the contribution of sensors. Under the sequential Bayesian filtering method, the information contribution of sensor k with the measurement $z_k^{(t+1)}$ is

$$I_{MI,k} = MI\left(X^{(t+1)}; Z_k^{(t+1)} \mid \overline{Z}^{(t)} = \overline{z}^{(t)}\right) = E_{p\left(x^{(t+1)}, z_k^{(t+1)} \mid \overline{z}^{(t)}\right)}\left[\log \frac{p\left(x^{(t+1)}, z_k^{(t+1)} \mid \overline{z}^{(t)}\right)}{p\left(x^{(t+1)} \mid \overline{z}^{(t)}\right) p\left(z_k^{(t+1)} \mid \overline{z}^{(t)}\right)}\right] \tag{2}$$

Intuitively, it indicates how much information $z_k^{(t+1)}$ conveys about the target location $x^{(t+1)}$ given the current belief. In the IDTR algorithm, we prefer to choose the node with the maximum mutual information as the next hop.

2.3 Related Works

For the scenario of routing the query from one proxy to the exit node with limited communication costs, [7] proposed an A* heuristic search algorithm. It selected four special paths as the representatives of the available paths that satisfied the requirement of the total communication costs. The path with maximum accumulated information gains was selected to perform sequential Bayesian filtering. The computation complexity may be beyond the capability of the small volume sensors, and the accumulated information gain can't be computed by simply summing up the values of several samples for its state dependency.

If sensor holes exist, [7] proposed a multiple step look-ahead algorithm. It evaluated the information contribution of all the nodes in M hop neighborhood, converted the graph representation of the network and tried to find the path with the maximum accumulated information gain. The assumption is that the sum of individual information $\sum I(v_k)$ can often be considered as a reasonable approximation of $I(v_1, \cdots, v_T)$ in cases where the belief state varies slowly. Because the neighbor sensors are close to each other, there is redundancy between $I(v_k)$ and $I(v_{k-1})$. So the assumption is not reasonable.

2.4 IDTR-Based Heuristic Search Algorithm

In this section, we introduce an IDTR-based heuristic search algorithm to improve the algorithm in [7]. It realizes locally optimization during the path selection.

2.4.1 Algorithm Description

The sensor on the path performs sequential Bayesian filtering by combining the received information content with its own measurement to estimate the target

location. When selecting the next hop, the proposed algorithm not only considers the information contributions of neighbors, but also the cost constraint C_0. The value of C_0 controls the tradeoff between the communication cost and the information aggregation. Smaller C_0 value favors shorter path, and larger C_0 allows longer path with more effective information aggregation.

Suppose that the path $P^{(t)}$ is planned up to the sensor v_t, $S^{(t)}$ is current belief at time t, and N_{v_t} is the list of neighbors within M hops to v_t. If the communication distance is used to represent the cost, in order to further arrive at the exit node v_{exit}, the remaining path length is upper bounded by $C = C_0 - C_{P^{(t)}}$, where $C_{P^{(t)}}$ is the communication length cost already paid. Our goal is to select the next hop whose information contribution is the largest under the constraint of the communication cost C. C confines the locus of all possible paths to form an ellipse which is defined by $|l - v_t| + |l - v_{exit}| = C$, where v_{exit} and v_t are the focuses.

Table 1. Algorithm for IDTR-based heuristic search at each sensor

C_0 : constraint of communication cost $P^{(t)}$: path planned up to time t $C_{P^{(t)}}$: communication length cost of path $P^{(t)}$ $S^{(t)}$: current belief at time t v_t : current active sensor performing path planning v_{exit} : exit node N_{v_t} list of neighbors within M hops to v_t
Step 0. Sensor v_t is idle until receive the query $\left(t, S^{(t)}\right)$ Step 1. Take new measurement $z^{(t+1)}$, update the belief to $S^{(t+1)}$ Step 2. Compute constraint on the remaining path length: $C = C_0 - C_{P^{(t)}}$ Step 3. If $C < L = \|v_t - v_{exit}\|$ Use the geographical routing algorithm; Else For $\forall v_j \in N_{v_t}$, Compute the distance summary $C_j = \|v_j - v_t\| + \|v_j - v_{exit}\|$ If $C_j < C$, Calculate its information contribution I_j Select the one with the maximum mutual information, $v_{\text{maxinfo}} = \arg\max_{k \in N} I_j$ Compute the shortest path from v_t to v_{maxinfo} using Dijkstra's algorithm Step 4. Forward the $\left(t+1, S^{(t+1)}\right)$ to v_{maxinfo}, and the middle sensors on the shortest path update the belief

Once receiving the routing request $\left(t, S^{(t)}\right)$, sensor v_t takes new measurement $z^{(t+1)}$, and updates the belief to $S^{(t+1)}$ through sequential Bayesian filtering. For $\forall v_j \in N_{v_t}$, sensor v_t calculates the sum of the distance, $C_j = \left|v_j - v_t\right| + \left|v_j - v_{exit}\right|$. $C_j < C$ means that v_j is within the range of the ellipse, and its information contribution needs to be calculated, assumed as I_j. The one with the maximum mutual information is selected as the next hop. Then construct a graph of v_t's M hop neighborhood, take the Euclidean distance as the weight of the edge between neighboring sensors, and compute the shortest path from v_t to v_{maxinfo} using Dijkstra's algorithm. The routing request $\left(t+1, S^{(t+1)}\right)$ is transferred to v_{maxinfo} along the shortest path. The middle sensors on the shortest path are only responsible for update the belief and transfer the belief state. The v_{maxinfo} will repeat this belief update and next hop selection process.

If there is no node within the range of the ellipse, select the sensor among N_{v_t} with the smallest C_j as the next hop. If the ellipse does not exist, the geographical routing algorithm is used to search for the shortest path. The IDTR-based heuristic search algorithm is summarized in Table.1.

2.4.2 Complexity Analyses

The computational complexity of the IDTR-based heuristic search algorithm is primarily determined by the computation of information utility function I. So we use the times of mutual information computation to represent the complexity. It depends on the constraints of the communication costs C_0, the average degree of the sensor D_s. If C_0 means the limited hop of the data transmission, then the total number of calculation is $N_{t1} = D_s \cdot C_0$.

For the algorithm proposed in [7], the computational complexity of estimating information is proportional to the product of the number of sampling paths N_p, the number of integral samples S_0, and the constraints of the communication costs. Hence, the overall complexity is about $N_{t2} = N_p \cdot S_0 \cdot C_0$. Assume that $D_s = 6$, $N_p = 4$ and $S_0 = 10$, N_{t2} is about 7 times as the N_{t1}.

3 Simulation Results and Analyses

Simulations are carried out to evaluate the performance of the proposed algorithm. We use the grayscale grids to represent the mass distribution of the target location. Blacker grid means that the target is more likely to be at the grid location. The initial belief is assumed to be uniformly distributed over the entire sensor field.

To measure the tracking performance, we consider two quantities: 1) the mean-squared error (MSE) $E_{p_{(x^{(t)}/\xi^{(t)})}}\left\|x^{(t)}-x_{true}^{(t)}\right\|^2$, which describes the locating accuracy; 2) the size of the belief, which is calculated as the number of cells with likelihood value larger than 0.001, reflecting the uncertainty.

3.1 Routing with the Communication Cost Constraints

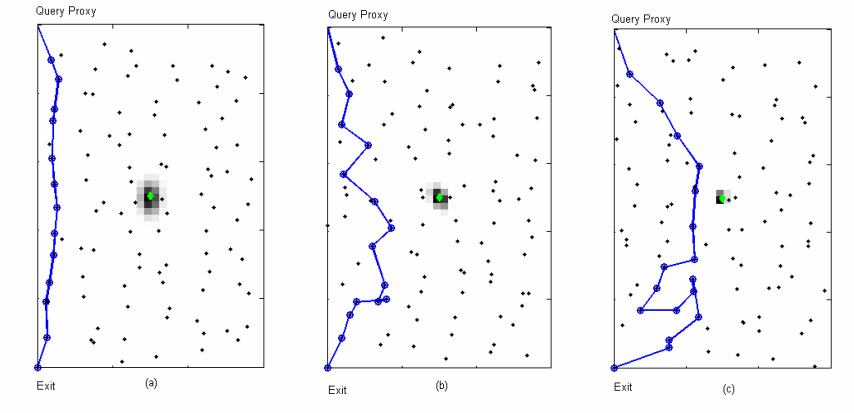
100 acoustic amplitude sensors[8] are uniformly distributed in the $150\times250\,m^2$ sensor field, and the range of radio is 30 m. The stationary target is located at $(75,125)$. We designate the sensor closest to the upper left corner as the query proxy node, and the sensor closest to the lower left corner as the exit node.

Our algorithm is essentially a tradeoff problem between the communication cost and the information aggregation controlled by the path length constraint C_0. We vary the allowance C_0 in $\{250,300,350,400,450\}$. For each value, 20 independent simulations are run to eliminate the effects of randomness. The statistical results are listed in Table 2. With the increase of C_0, the average number of hops increases also, while both the MSE and the belief size decreases. Compared with the shortest path scenario, the IDTR-based heuristic search algorithm takes a little bit of detour, but considerably improves the locating performance.

Table 2. Performance Comparison with different C_0 values

C_0	MSE	Belief size	Number of Hops
Shortest Path	6.8047	34.0	12.0
300	5.3957	22.7	13.6
350	4.8295	18.1	15.8
400	4.0844	14.4	18.0
450	3.7511	12.2	22.8

Fig.4 visualizes the selected paths and the grayscale graph of the target location under different cost constraints. Fig.4(a) is the shortest path scenario. The path contains 13 hops and mostly follows a vertical line from the query proxy to the exit node. The final location grayscale graph is a little bit dispersive. Fig.4(b) shows a longer path with 15 hops. Starting from the query proxy, the path bends towards the target direction and attempts to accumulate more information. The tracking performance is vastly better than that of the shortest path scenario. Fig.4(c) shows a path with 17 hops. The tracking accuracy is further improved.



(a) MSE=7.0441, belief size=35 (b) MSE=4.8163, belief size=16 (c) MSE=2.9026, belief size=6

Fig. 4. Path selection and target location grayscale graph under different C_0

3.2 Routing with Sensor Holes

We present simulation results for routing a query from an arbitrary query proxy node to high information content region with sensor holes in the network. Sensor layout is generated as follows: generate a uniform grid points of 15 rows and 6 columns to evenly cover the region, and then perturb the grid points with independent Gaussian noise of $N(0,5)$. To test the routing performance in the presence of sensor holes, we remove the points in row 5 and 6 and columns 2-5, resulting a 4×2 routing hole. We simulate the task of tracking moving target, which moves along the straight line $x=75$ with speed $v=7m/s$.

Fig.5 shows several snapshots of the routing path applying our routing algorithm, which represent the tracking scenarios at 20, 25 and 30 hops respectively. From the figures we can see that in the IDTR-based heuristic search algorithm, because the current sensor possesses the information of all M hop neighbors when selecting node, the route can bypass the routing hole and extend to the target location, indicating that the target locating is successful.

In Fig.5(a), even the target falls into the sensor holes, our algorithm still can bypass the holes successfully, and collect information as more as possible around the target to locate it. The true location is $(75.0,106.0)$, and the estimated position is $(75.3,107.2)$. In Fig.5(b), the path moving along with the target, occasionally reach neighborhood to gather information. The true location is $(75.0\ \ 141.0)$ and the estimated position is $(77.1\ \ 142.4)$. In Fig.5(c), the true location is $(75.0\ \ 176.0)$ and the estimated position is. All realizes the object of correctly tracking.

The performance of our algorithm with the one proposed in [7] is compared. As illustrated in Fig.6 (b), in order to arrive at the sensor with the maximum information

contribution, the path visits several middle sensors and combines their measurements to refine the estimated belief. So the cloud is more compact. The numerical values are listed in Table 3. Our algorithm achieves better target locating performance with the cost of more communication hops.

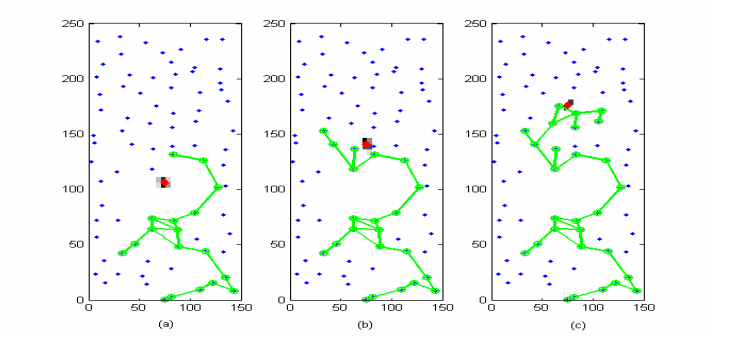


Fig. 5. Tracking moving target

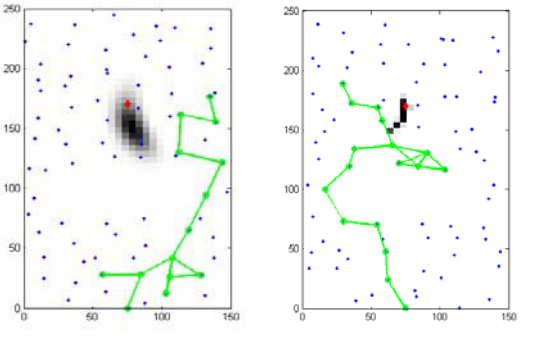


Fig. 6. Two snapshot of the multiple step look-ahead approach (a) and the IDTR-based heuristic search method (b)

Table 3. Performance comparision of the two algorithms

	Multiple Step Look-Ahead Approach [8]	IDTR-based Heuristic Search Algorithm
Number of hops	16	26
Estimated Location	(80.226, 153.32)	(73.0719,163.0944)
Belief Variance	33.564	8.158
Error Distance	17.478	7.170
Belief Size	89	14

4 Conclusions

Tracking problem is one of the most important applications for WSNs. This paper adopts information-driven task routing algorithm to solve the special routing scenario with communication cost constraints and sensor holes. Sensors on the path update the target belief via sequential Bayesian filtering, evaluate the information utility of M hop neighbor sensors as well as the distance between the neighbors and the exit node, and jointly optimize the selection of the next hop. The main goals of the proposed algorithm are to minimize the communication cost and simultaneously to maximize the accumulated information gain. In the future, we will apply the IDTR algorithm in the tracking problem of several moving targets and the target estimation problem.

References

1. Shah, R.C., Rabaey, J.M.: Energy aware routing for low energy ad hoc sensor networks. In Proc. IEEE Wireless Commun. Netw. Conf., Orlando, FL, (2001) 350-355
2. Intanagonwiwat, C. Govindan, R. Estrin, D. Heidemann, J. Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on Networking. Volume 11, issue 1, (2003) 2-16
3. Stojmenovic, I., Lin, X.: Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. IEEE Trans. Parallel Distrib. Syst., Volume 12, No. 10, (2001) 1023-1032.
4. Karp, B., Kung, H.T.: Greedy perimeter stateless routing for wireless networks. In Proc. MobiCom, Boston, MA, (2000) 243-254
5. Manyika, J., Durrant-Whyte, H.: Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach. Ellis Horwood, New York (1994)
6. Feng, Zhao, Jaewon, Shin, Reich, J.: Information-driven dynamic sensor collaboration. IEEE Signal Processing Magazine, Volume 19, Issue 2, (2002) 61-72
7. Liu, J., Feng Zhao, Petrovic, D.: Information-directed routing in ad hoc sensor networks. IEEE Journal on Selected Areas in Communications, Volume 23, Issue 4, (2005) 851-861.
8. Juan. Liu, Reich, J.E., Feng, Zhao: Collaborative in-network processing for target tracking. EURASIP, J. Appl. Signal Process, vol. 2003, Mar. (2003) 378-391
9. Haykin, S.: Adaptive Filter Theory. Prentice Hall Inc., 4th Ed., 2002
10. M., Chu, H., Haussecker, Feng, Zhao: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. Int. J. High-Performance Comput. Appl., vol. 16, no. 3. (2002) 293-313
11. T.M., Cover, J.A., Thomas: Elements of Information Theory. Wiley, New York, (1991)

Autonomic Management of Scalable Load-Balancing for Ubiquitous Networks

Toshio Tonouchi and Yasuyuki Beppu

Internet Systems Laboratories, NEC Corporation
{tonouchi@cw, y-beppu@ak}.jp.nec.com

Abstract. In ubiquitous networks, a lot of sensors and RFID readers will be connected to the networks as well as PCs and mobile phones are connected, and huge numbers of transactions are expected to be issued by these devices. Since the loads caused by the transactions are getting increased gradually, it is difficult for system manager to estimate the required performance. The scalable load balancing method is, therefore, expected. This report proposes the scalable load-balancing method in which servers communicate each other, and they manage themselves for load-balancing. Because this method does not need a center server, it can avoid the center-server bottleneck. This system, however, is prone to diverge; the system continues load balancing repeatedly in certain configuration and never stops load-balancing operations. In this paper, we clarify a sufficient condition of the divergence, and prevent the system from the divergence.

Keywords: load balancing, autonomic algorithm, ubiquitous network.

1 Introduction

A lot of traffic is expected to be issued by a huge number of PCs and cellular phones as well as by RFID readers and sensors in ubiquitous networks. For example, all merchandises and products will be attached with RFID tags, and they may be traced, in whole supply chains, by RFID readers equipped by warehouses, trucks, smart shelves, shopping carts. Another example is sensor systems for secure societies. Sensors and surveillance cameras, like CCTV in UK, are equipped at every place in cities. Logs of sensors and videos which the cameras take come and go through ubiquitous networks. We assume that in near future from a million to ten million sensors and RFID readers are allocated everywhere. These sensors and readers are assumed to sense targets once a second, and, as a result, 1 [request/sec] × 10 million [sensors] = 10million [request/sec] of requests are issued. The traffic is 100 GB/sec if an average size of packet is about 1kB. Fig. 1 shows the performance of local load balancers [1]. This graph shows that they cannot handle 100GB/sec of packets whose size is 1kB. New load balancing methods are, therefore, required for the ubiquitous era.

Our target system is a web system, which is composed of clients, AP servers and a database system. We put the following assumption on the target system.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 33 – 42, 2006.
© Springer-Verlag Berlin Heidelberg 2006

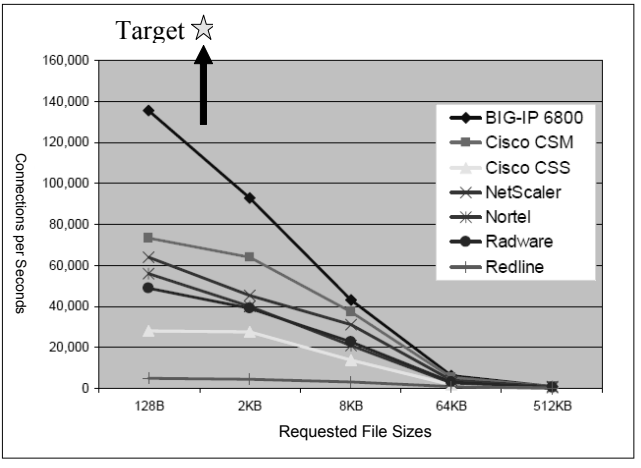


Fig. 1. Performances of local load balancers [1]

- AP servers should be stateless. In another word, the states of applications running on the AP servers are stored in the database system. For example, AP servers retrieve the states of application from the database system with the key which is encoded in cookies in clients.
- The database system supports a transaction mechanism. AP servers can use the mechanism and they can store and retrieve data from the database system atomically. In another word, a store or retrieval operation of AP servers caused by an request from a client may succeed, or failed. When failed, the operation is canceled and no effect is applied to the database. In this paper, we only discuss robustness of the AP servers, and we do not handle the robustness of database servers. It is a future work.
- The protocol between servers and clients (clients may be edge servers connected to RFID readers or sensors) supports a redirection mechanism. The mechanism enables servers to indicate clients to access another server. An example of a protocol supporting is HTTP [2]. In HTTP, when a servers reply to client with 302 response code, the client redirects to the server which is suggested with the 302 response.
- The AP servers, the database system and the clients are distributed to whole network, and they are connected.

In Section 2, previous load balancing methods are given. We show our algorithm in Section 3. Section 3 also includes preliminary experiments, which shows a problem of the proposed method. We clarify the mechanism causing the problem and show how to prevent from the problem. Section 4 shows by experimental evaluations the prevention succeeds.

The contributions of this paper are as follows:

- We proposed a distributed load balancing mechanism without a center server. We can, therefore, avoid bottleneck of the center server.
- No center server mechanisms, like the proposed method, inherently fall into divergence. For example, servers continue trying to distribute the load to other servers, but the distribution does never stop and, in addition, some server may receive more load than ever. We show that the divergence in experiments, and we clarify that the divergence mechanism by mathematical model. The model clarifies a sufficient condition that the method converges. The system is guaranteed to converge if we design the system obeying the sufficient condition.

2 Related Work

A lot of valuable work has been done in the load balancing techniques. These works are sorted into two types: one is a local load balancing technique and the other is a global load balancing technique. Many vendors provide many kinds of local load balancers. A local load balancer has a virtual IP address which is open to clients. The load balancer dispatches incoming requests from the virtual IP address to several back-end servers. It is easy to introduce local load balancers because an existing server can be replaced with a load balancer and its back-end servers. In addition, the system with a local load balancer can manage the back-end server failure because a load balancer can easily monitor its back-end servers. However, a load balancer itself can be bottleneck when huge traffics come. It is because a local load balancer has only one virtual IP address and it must handle all traffics. As a result, local load balancing technique has limitation in scalability.

A typical method of global load balancing techniques is a DNS round-robin method; a special DNS server can be placed in the network (Fig. 2). The DNS server answers the IP address of a not-busy server or a nearest server to a client which sends a DNS request. The client, therefore, can access the not-busy server which is expected to reply quickly or the nearest server which can communicate to the client with a small network delay. However, caches of DNS answers result in the inaccurate load balancing [3]. DNS servers consist of the tree structures whose roots are thirteen root DNS servers in the world. A local DNS server may have caches of the answers from its parent DNS server. The local DNS server may not know the change of the load balancing situation because an “active” cache is used instead of the information of its parent DNS server. To avoid cache problem, only one DNS server may gather all the load information, and all the clients may access the DNS server. In this method, however, the bottleneck of the DNS server emerges. It is said that the DNS server can handle about thousands of requests per seconds. Because each server access of a client accompanies with a DNS request, the DNS server is expected to handles 1million – 10million [request/sec], but it may be impossible.

f5 3DNS system [4], which is now called Global Traffic Manager, tries to solve the cache problem. 3DNS servers are allocated as local DNS servers, and they always

refresh load-balancing information with a proprietary protocol. Fig. 3 shows the mechanism of the 3DNS system. Each local network is equipped with a 3DNS server. When a user belonging to a local network accesses a server with URL, e.g. http://newyork.myFirm.com/, his/her client host asks the 3DNS server in the local network the IP address corresponding to the URL (Step 1 in Fig. 3). The local DNS server receives the query, and it also asks a 3DNS server (Step 2). The 3DNS server asks backend servers distributed among network systems its load and the network distances between the 3DNS server and the backend servers (Step 3). It is assumed that each local network is equipped with the f5 load balancers. The load information and the network distances are acquired with the proprietary protocol between a 3DNS server and f5 load balancers. The 3DNS server answers to the client the IP address of an adequate server, considering the load information and the network distances (step 4).

Because only clients in each local network can access a 3DNS server, the performance of single 3DNS server need not be excellent. The 3DNS system, however, can be scalable. In addition, proprietary protocol can reduce the cache problem. However, the 3DNS system assumes that each local network must is equipped with a 3DNS server, and f5 load balancers. The system is difficult to be deployed.

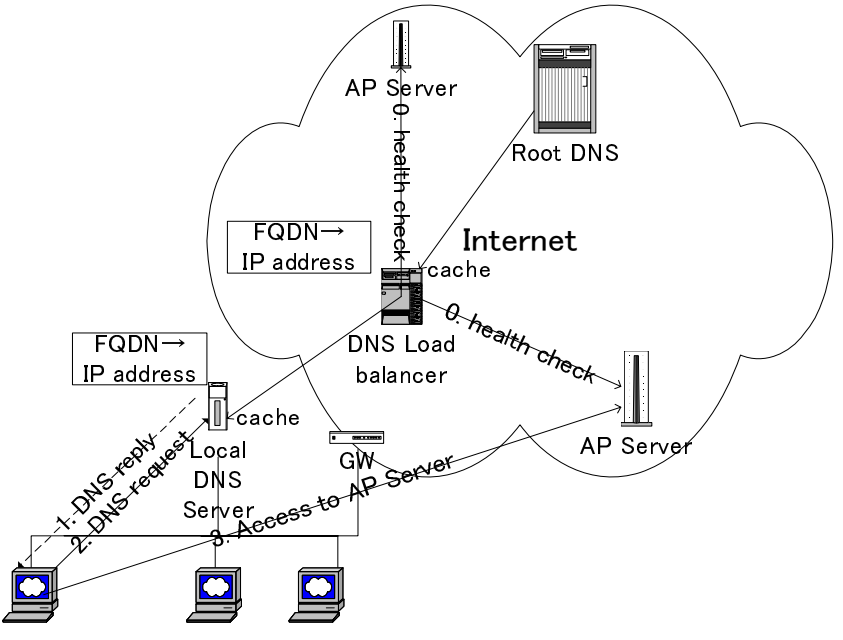


Fig. 2. DNS Round-robin technique

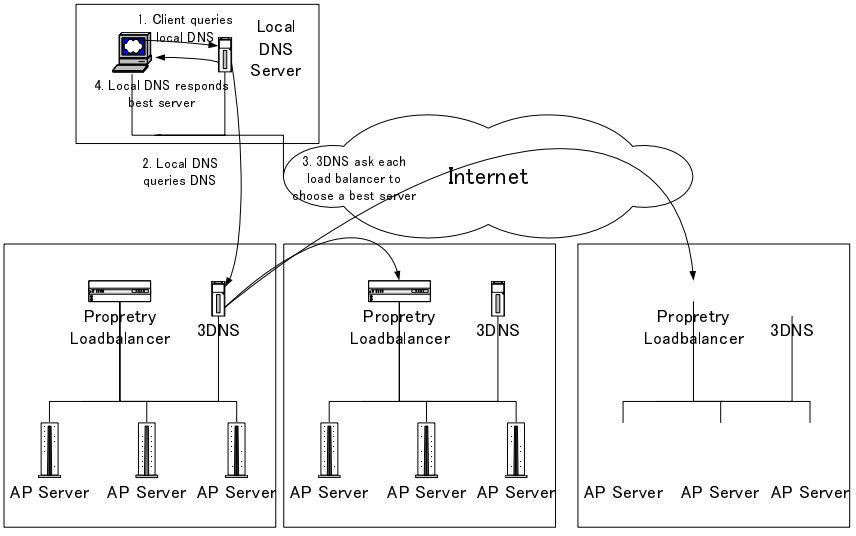


Fig. 3. Mechanism of 3DNS [4]

There are some activities in which Distributed hash table technique (DHT) is used as a global load balancing mechanism. In short, jobs or contents are assigned to servers in global networks by DHT. Jobs or contents are assigned the servers of IDs same with the ID which a hash function applied to the jobs or contents calculated. This may result in load balancing because the assignment by the hash function seems to be random. Byes points out that this mechanism can not be a fair load balancing [5]. Fig. 4 shows the problem when Chord [6] is used. Chord is not guaranteed to assign servers into a "ring" in equal distances. A server whose ring distance between itself and a "previous" server is long may be assigned more jobs.

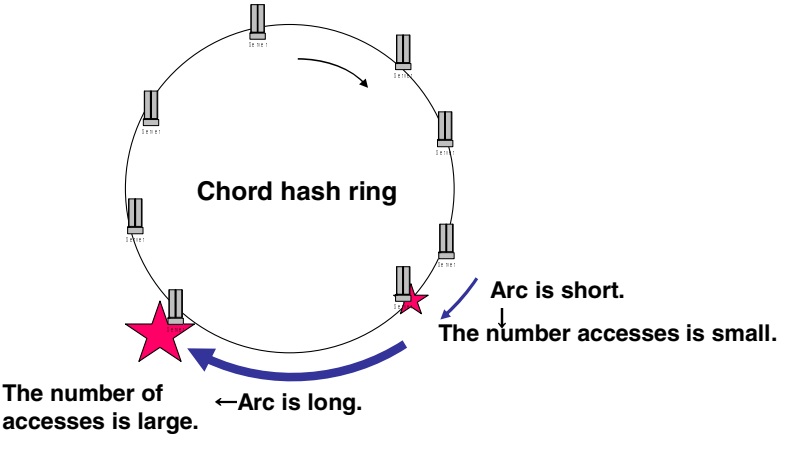


Fig. 4. Unfair load balancing with Chord

3 Approach: Load Exchange Method

We provide a naïve scalable load balancing method called a *load exchange method*. It is scalable because no center management server exists. We can avoid so-called a center-server bottleneck. No-center approaches usually result in divergence; the algorithm does not stop and result in worse situation. Our preliminary experiments show that our algorithm is guaranteed of convergence; the algorism always stops and results in almost equally load-balanced saturation.

3.1 Algorithm

Fig. 5 shows the intuitive image of the load exchange method. We assume that servers know some servers (we call them neighbors), and this makes a server graph $G = (V, E)$, where V is a set of servers, and $E : (V, V)$ is a set of a pair of a server and its neighbor. We call E a set of edges. We assume that the number of edges connected to a server is less than a constant number (e.g. 5) even if the number of N becomes big. In our algorithm mentioned later, each server communicates with its neighbors, and a large number of neighbors may become overhead.

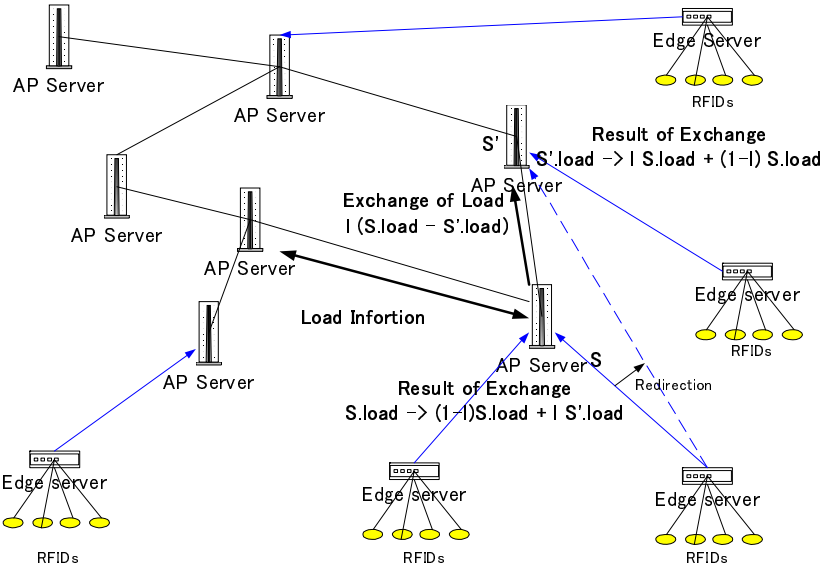


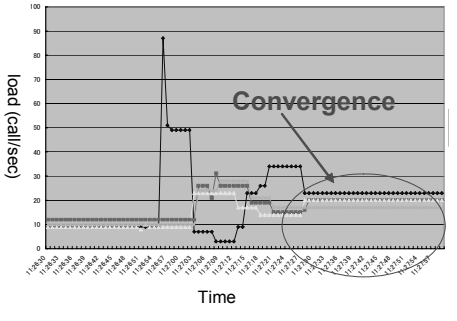
Fig. 5. Overview of the load exchanging method

"S.load" means the load of Server $S \in V$. A load may be a number of requests per second, a CPU load, a response time, and so on. Server S and its neighbor S' tell each other their load information periodically. The number of neighbor servers is less than given small constant. It means that the overhead and delay issued by this communications are little because each server communicates with a small number of neighbors.

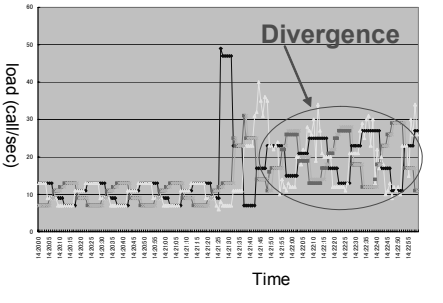
Sever S gives S' some jobs when S.load / S'.load > D, where D is a given *load exchange threshold* (D≧1). In this case, S gives its job to S' so that the load of S becomes "S.load – *l* (S.load - S'.load) = (1-*l*) S.load + *l* S'.load" and so that the load of S' becomes "S'.load + *l* (S.load - S'.load) = (1-*l*) S'.load + *l* S.load", where *l* is a given *load exchange factor* (0 < *l* < 1). The load exchange is realized with redirection mechanism. For example, in HTTP case, Server S issues 302 response with the URL of Server S'. The client re-issues the request to S' when it receives the response. Server S repeats to answer a 302 response until the load becomes "(1-*l*) S.load + S'.load".

3.2 Problem of the Load Exchange Method

Fig. 6 is a preliminary evaluation result of the load exchange method. We implemented the method over four TOMCAT [7] AP servers. Fig. 6 is a graph of the load balancing among these four servers. The x-axis shows time, and the y-axis shows the loads of the four servers. The left graph is evaluated under *l* = 0.6, and the right one is under *l* = 0.8. One hand, the left one shows that all loads of the servers converge and the all loads of servers become almost equal. On the other hand, the right one shows divergence; the loads do not result in a stable state. We call it a divergence problem.



Conditions: Complete graph of four nodes,
D = 2, l = 0.6



Conditions: Complete graph of four nodes,
D = 1.2, l = 0.8

Fig. 6. Preliminary evaluation results of load exchange method

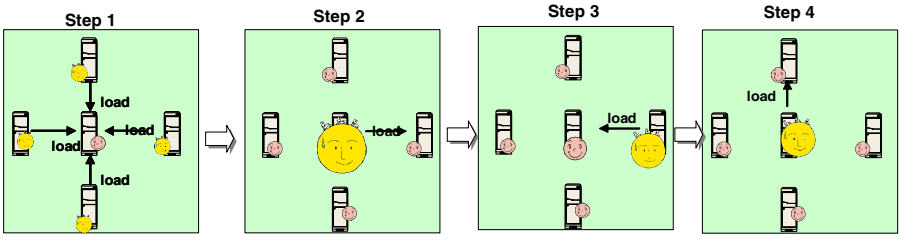


Fig. 7. Mechanism for the divergence problem

Fig. 7 shows the reason of the divergence. When a server with a light load (at the center of the figure) is connected to servers with a heavy load (step 1), the server with light load receives the load from its connected servers. As a result, it must manage heavy load (Step 2). It, then, gives its load into another connected server (e.g. a server in right side) (Step 3). And the right side server will try to give its load into another server in the next time (Step 4).

3.3 Mathematical Models and Sufficient Condition of Convergence

It is important to guarantee the load exchange algorithm to converge. We clarify the behavior of the load exchange algorithm with a mathematical model, and find a sufficient condition.

In the model, we ignore Load balancing threshold D because the simplicity of the model. In another word, we consider D be 1. The algorithm converges under D > 1 if the algorithm converges under D = 1. Therefore, the convergence condition under D = 1 is a sufficient condition to those under $D \geqq 1$.

We have n servers : $0, \cdots, (n-1)$. $x_{i,t}$ is the load of Server i on Time t and $\vec{x}_t \overset{def}{=} {}^t\left(x_{0,t} \cdots x_{n-1,t}\right)$ where t means a transpose matrix. The load exchange is modeled by :

$$x_{i,t+1} = x_{i,t} - l \sum_{j \in N(i)} (x_{i,t} - x_{j,t}) \tag{1}$$

where $N(i)$ is a set of servers connected to server i. In short, there is $N(i) \overset{def}{=} \{ j \in \mathrm{N} \mid (i,j) \in \mathrm{E} \}$ where $\mathrm{G} = (\mathrm{V}, \mathrm{E})$. It is because Server i gives Server j Load $-l(x_{i,t} - x_{j,t})$ when $x_{i,t} > x_{j,t}$. We define Adjacency matrix $\mathbf{A} \overset{def}{=} \begin{pmatrix} a_{0,0} \cdots a_{0,n-1} \\ \vdots \quad \ddots \quad \vdots \\ a_{n-1,0} \cdots a_{n-1,n-1} \end{pmatrix}$ where $a_{i,j} \overset{def}{=} \begin{cases} |N(i)| & \textbf{if } i = j \\ -1 & \textbf{if } j \in N(i) \\ 0 & \textbf{if } j \notin N(i) \end{cases}$. Notice that

$$\sum_{j=0}^{n-1} a_{i,j} = 0 \tag{2}$$

because the number of neighbors equals to the sum of edges. We can express (1) as

$$\begin{aligned} \vec{x}_{t+1} &= (-lA + E)\vec{x}_i \\ \vec{x}_t &= (-lA + E)^{t-1}\vec{x}_0 \end{aligned} \tag{3}$$

where E is a unit matrix.

It is known that $\lim_{t\to\infty} x_{i,t}$ converge if all eigen values of $\mathbf{H} \overset{def}{=} (-l\mathbf{A} + \mathbf{E})$ are larger than -1, and 1 or below. It is clear that $\det(\mathbf{H} - \alpha\mathbf{E}) = 0$ where Eigen value α of $\mathbf{H}$. One of the eigen values of $\mathbf{G}$ is 1 because $\det(\mathbf{H} - \alpha\mathbf{E}) = \det(\mathbf{H} - \mathbf{E}) = \det(-l\mathbf{A}) = l^n \det(\mathbf{A})$ if $\alpha = 1$. And then, you can easily find $\det(\mathbf{A}) = 0$ because of (2). In addition, the eigen vector corresponding to Engen value $\alpha = 1$ is $\vec{\alpha}_i = {}^t(1 \cdots 1)$ because $\mathbf{H}\vec{\alpha}_i = \vec{\alpha}_i$. This eigen vector means that all loads are equalized in future. In short, the sufficient condition of the convergence is that all the eigen values α of $\mathbf{H}$ are $-1 < \alpha \le 1$. All loads converge in the equal loads if the sufficient condition is satisfied.

4 Experimental Evaluations

In order to prove the correctness of the mathematical model, we compare the simulation results based on the mathematical model with the experimental evaluations. We give burst requests to a server, and Fig. 8 shows the loads of servers after the burst requests. The upper three graphs are the results of the simulations and the lower three graphs are those of the experimental results. Each pair of graphs in the same row is evaluated under the same condition. The left pair is evaluated under $l = 0.4$, the center pair is evaluated under $l = 0.6$, and the right pair is evaluated under $l = 0.8$. Four servers are connected to the other servers in all evaluations. In short, servers in each evaluation make a complete graph.

Both graphs in the left pair converge because the max absolute value of eigen values except 1 is -0.84 in the left pair. Both of the graphs in the right pair diverge because the max absolute value of eigen values is -2.7 in the right pair. These two

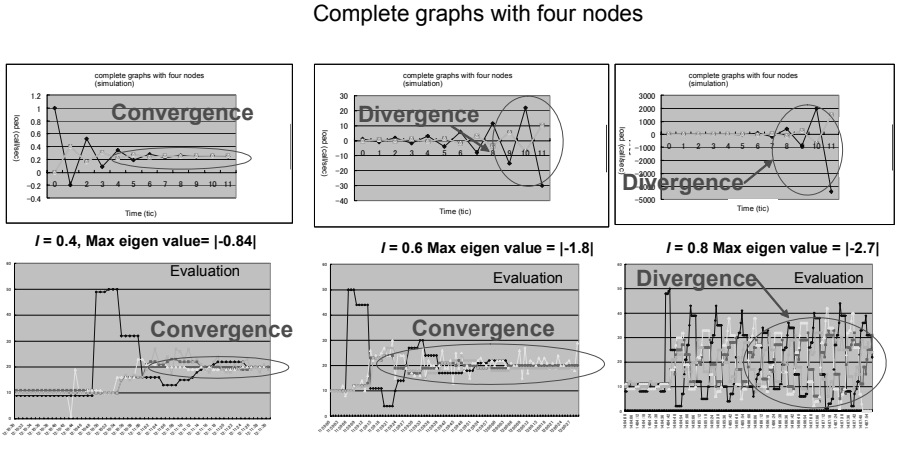


Fig. 8. Simulations based on the mathematical model and experimental evaluations

results show that the model is correct in the experiments. Notice that the lower graph in the center pair converges while the upper one diverges. We think that the convergence condition given in Section 0 is sufficient condition. Therefore, there some cases where the system converges while the corresponding model divergences.

5 Conclusion

We propose the naïve global load balancing method in this paper. The method is prone to diverge, but we model the method and clarify the sufficient condition that the load balancing converges. The correctness of the model is proved through the experimental result.

This algorithm assumes that Graph G is given. We are now developing the autonomic graph making protocols. We aim at the maintenance-free and fault-tolerant graph construction.

Acknowledgement

This research was sponsored by the Ministry of internal Affairs and Communication in Japan.

References

1. F5, *Broadband-Testing: Application Traffic Management* , Jan., 2005
2. T. Berners-Lee, et al., *Hypertext Transfer Protocol -- HTTP/1.0*, RFC 1945, May 1996
3. Tony Bourke, *Server Load Balancing*, O'reilly, 2001, ISBN 0-596-00050-2
4. f5, "BIG-IP Global Traffic Manager", http://www.f5.com/products/bigip/gtm/
5. Byes, J., et al. *Simple Load Balancing for Distributed Hash Tables*, in Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), pp. 80-87
6. Stoica, I., et al. *Chord: A scalable peer-to-peer lookup service for internet applications,* In ASM SIGCOMM 2001, pp.149-160
7. The Apache Software Foundation, *Apache Tomcat*, http://tomcat.apache.org/

A Policy-Based Management Framework for Self-managed Wireless Sensor Networks*

Jong-Eon Lee[1], Si-Ho Cha[2,**], Jae-Oh Lee[3],
Seok-Joong Kang[1], and Kuk-Hyun Cho[1]

[1] Department of Computer Science, Kwangwoon University, Seoul, Korea
{jelee, sjkang, khcho}@cs.kw.ac.kr
[2] Department of Computer Engineering, Sejong University, Seoul, Korea
sihoc@sejong.ac.kr
[3] Department of Information and Communication, KUT, Chungnam, Korea
jolee@kut.ac.kr

Abstract. This paper proposes a policy-based management framework for self-managed wireless sensor networks (WSNs) called SNOWMAN (SeNsOr netWork MANagement). In WSNs, a number of sensor nodes are deployed over a large area and long distances and multi-hop communication is required between nodes. So managing numerous wireless sensor nodes directly is very complex and is not efficient. The management of WSNs must be autonomic with a minimum of human interference, and robust to changes in network states. To do this, our SNOWMAN architecture is based on hierarchical management architecture and on the policy-based network management (PBNM) paradigm. SNOWMAN can reduce the costs of managing sensor nodes and of the communication among them using hierarchical clustering architecture. SNOWMAN can also provide administrators with a solution to simplify and automate the management of WSNs using PBNM paradigm.

1 Introduction

In Wireless sensor networks (WSNs), radio bandwidth is scarce, computational power is limited, and energy efficient is paramount. Such limitations are challenges to overcome. In particular, one of the essential needs is for a system that autonomously manages the limited energy and bandwidth of WSNs. In WSNs, a number of sensor nodes are deployed over a large area and long distances and multi-hop communication is required between nodes and sensor nodes have the physical restrictions in particular energy and bandwidth restrictions. So managing numerous wireless sensor nodes directly is very complex and is not efficient. To make sensor nodes perform intelligent self-management, they should be organized and managed automatically and dynamic adjustments need to be done to handle changes in the environment. In [1], we propose an autonomous

* The present research has been conducted by the Research Grant of Kwangwoon University in 2006.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 43–52, 2006.
© Springer-Verlag Berlin Heidelberg 2006

architecture for WSNs called SNOWMAN (SeNsOr netWork MANagement), which is based on policy-based network management (PBNM) paradigm and hierarchical clustering architecture. PBNM paradigm of SNOWMAN can provide administrators with a solution to simplify and automate the management of WSNs. However, the cost of PBNM paradigm can be expensive to some WSN architecture. SNOWMAN employs therefore hierarchical clustering management architecture, which can reduce the costs of managing sensor nodes and of the communication among them, using clustering mechanisms.

This paper is structured as follows. Section 2 investigates related researches. Section 3 discusses the architecture and components of the proposed SNOWMAN. Section 4 evaluates our hierarchical clustering algorithm. Section 5 presents the implementation of the SNOWMAN. Finally in section 6 we conclude the paper.

2 Backgrounds

2.1 Management Architectures

Linnyer B. Ruiz designed the MANNA architecture [4] for WSNs, which considers three management dimensions: functional areas, management levels, and WSN functionalities. He also proposed WSN models to guide the management activities and the use of correlation in the WSN management. However, he described only conceptual view of the distribution of management functionalities in the network among manager and agent.

Mohamed Younis [5] proposed architecture for monitoring and management of sensor networks, which focuses on reducing the sensitivity of the operation and monitoring of sensor networks to the ambiguity of the propagation model of the radio signal. He suggested agent sensors that relay messages to and from unreachable sensors and groups of sensors around these agents while considering the load on each agent.

Chien-An Lee [6] proposed an intelligent self-organization management mechanism for sensor networks. The nodes are classified into three levels according to their functionality. The nodes in the low level are managed by those in the higher level and form hierarchical management structures. His work indicates how high-level nodes form a cluster through a contest with low-level nodes. Performance measures the cover loss, the average delay between the header and the member nodes and the 20∼80 rules are also considered.

2.2 Clustering Algorithms

To improve the clustering, several clustering algorithms have been proposed. Noted two schemes are LEACH and LEACH-C.

LEACH (Low Energy Adaptive Clustering Hierarchy). LEACH includes distributed cluster formation, local processing to reduce global communication, and randomized rotation of the cluster heads. These features leads a balanced energy consumption of all nodes and hence to a longer lifetime of the network [7]. However, LEACH offers no guarantee about the placement and number of cluster head nodes.

LEACH-C (LEACH-Centralized). Using a central control algorithm to form the clusters may produce better clusters by dispersing the cluster head nodes throughout the network. This is the basis for LEACH-C, a protocol that uses a centralized clustering algorithm and the same steady-state protocol as LEACH. Therefore the base station determines cluster heads based on nodes' location information and energy level. This feature leads to organize robust clustering topology [8]. However, frequent communications between the base station and other sensor nodes increases communication cost.

3 SNOWMAN (SeNsOr netWork MANagement)

3.1 Overview of Architecture

To facilitate scalable and localizable management of sensor networks, SNOWMAN constructs 3 tier regional hierarchial cluster-based senor network: regions, clusters, and sensor nodes as shown in Fig. 1.

In the architecture, a WSN is comprised of a few regions and a region covers many clusters which have several cluster head nodes. Sensor nodes should be aggregated to form clusters based on their power levels and proximity. That is, a subset of sensor nodes are elected as cluster heads. In 3 tier regional hierarchical architecture of SNOWMAN, cluster heads constitute the routing infrastructure, and aggregate, fuse, and filter data from their neighboring common sensor nodes. The PA can deploy specific policies into particular areas (or clusters) to manage just singular regions or phenomena by more scalable manner. So, SNOWMAN architecture is very useful to regionally manage the sensor networks.

Our SNOWMAN architecture includes a policy manager (PM), one or more policy agent (PAs) and a large number of policy enforcers (PEs) as shown in Fig. 1. The PM is used by an administrator to input different policies, and is located in a manager node. A policy in this context is a set of rules that assigns management actions to sensor node states. The PA and the PE reside in the base station and in the sensor node, respectively. The PA is responsible for interpreting the policies and sending them to the PE. The enforcement of rules on sensor nodes is handled by the PE. In a WSN, individual nodes will not be able to maintain a global view of the network. Such a task is well suited for a machine not constrained by battery or memory. This is the reason for having the PA on the base station.

It is the job of the PA to maintain this global view, allowing it to react to larger scale changes in the network and install new policies to reallocate policies (rules). If node states are changed or the current state matches any rule, the PE performs the corresponding local decisions based on local rules rather than sends information to base station repeatedly. Such policy execution can be done efficiently with limited computing resources of the sensor node. It is well known that communicating 1 bit over the wireless medium at short ranges consumes far more energy than processing that bit.

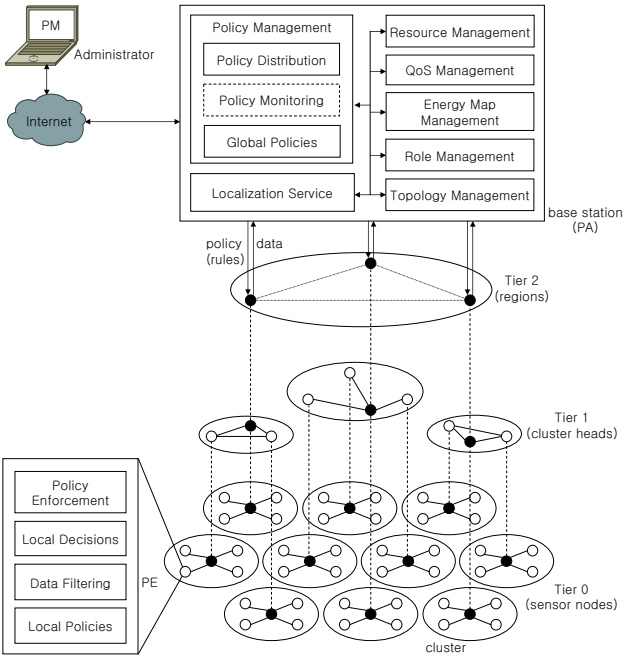


Fig. 1. SNOWMAN Architecture

3.2 Functional Components of SNOWMAN

The PA consists of several functional components: policy distribution, policy monitoring, resource management, energy map management, QoS management, topology management, role management, and localization service. Global policies are specified by a network administrator in a logically centralized fashion.

Policy distribution is the first essential task in ensuring that nodes are managed consistently with the defined policies. We design and implement a TinyCOPS-PR protocol that is similar to COPS-PR [9] protocol to deploy policies into sensor nodes. COPS-PR protocol is an extension for the COPS protocol to provide an efficient and reliable means of provisioning policies. The PA communicates with the PE using the TinyCOPS-PR protocol to policy distribution. TinyCOPS-PR allows asynchronous communication between the PA and the PEs, with notifications (reports, changes in policies, etc.) conveyed only when required.

Energy map management continuously updates the residual energy levels of sensor nodes, especially of cluster heads and region nodes. This energy map management is also achieved via topology management process. Topology management consists of a topology discovery, resource discovery, and role discovery. Resource management and role management manage the detected resources and roles, respectively. QoS management is a part of policy management using QoS policies like bandwidth allocation for emergency. Energy map management

and/or QoS management go through an aggregation and fusion phase when energy and/or QoS information collected are merged and fused into energy and/or QoS contours by means of cluster heads.

The PE enforces local policies assigned by the PM to make local decisions and filter off unessential redundant sensed data. To do this, the PE consists of policy enforcement function, local decision function, data filtering function, and local policies. The PE communicates with the PA via TinyCOPS-PR protocol to be assigned local policies.

3.3 SNOWcluster Algorithm

We propose a clustering scheme solely from a management viewpoint. Each sensor node autonomously elects cluster heads based on a probability that depends on its residual energy level. The role of a cluster head is rotated among nodes to achieve load balancing and prolong the lifetime of every individual sensor node. To do this, SNOWMAN re-clusters periodically to re-elect cluster heads that are richer in residual energy level, compared to the other nodes. We assume all sensor nodes are stationary, and have knowledge of their locations.

SNOWMAN constructs hierarchial cluster-based senor network using SNOW cluster clustering algorithm. SNOWcluster takes a couple of steps to accomplish the hierarchial clustering: 1) cluster head selection and 2) region node selection. In order to elect cluster heads, each node periodically broadcasts a discovery message that contains its node ID, its cluster ID, and its remaining energy level. After forming clusters, region nodes are elected from the cluster heads.

A node declares itself as a cluster head if it has the biggest residual energy level of all its neighbor nodes, breaking ties by node ID. Each node can independently make this decision based on exchanged discovery messages. Each node sets its cluster ID (c_id) to be the node ID (n_id) of its cluster head (c_head). If a node i hears from a node j with a bigger residual energy level (e_level) than itself, node i sends a message to node j requesting to join the cluster of node j. If node j already has resigned as a cluster bead itself, node j returns a rejection; otherwise node j returns a confirmation. When node i receives the confirmation, node i resigns as a cluster head and sets its cluster ID to node j's node ID.

When the cluster head selection is completed, the entire network is divided into a number of clusters. A cluster is defined as a subset of nodes that are mutually reachable in at most 2 hops. A cluster can be viewed as a circle around the cluster head with the radius equal to the radio transmission range of the cluster head. Each cluster is identified by one cluster head, a node that can reach all nodes in the cluster in 1 hop.

After the cluster heads are selected, The PA should select the region nodes in the cluster heads. The PA receives cluster information messages (c_info_msgs) that contain cluster ID, the list of nodes in the cluster, residual energy level, and location data from all cluster heads. The PA suitably selects region nodes according to residual energy level and location data of cluster heads. If a cluster head k receives region decision messages (r_dec_msgs) from the PA, the node k

compares its node ID with the region ID (r_id) from the messages. If the previous comparison is true, node k declares itself as a region node (r_node) and sets its region ID to its node ID. Otherwise, if node k's node ID is included in a special region list (r_list) from the message, node k sets its region ID to a corresponding region ID of the message. The region node selection is completed with region confirmation messages (r_conf_msgs) broadcasted from all of cluster heads.

4 Evaluation of SNOWcluster Algorithm

Each experiment is conducted via each of Leach, Leach-C and SNOWcluster clustering techniques. In addition, management messages are applied for all cases and the processing power of sensor nodes is eliminated because it is insignificant compared to the amount of energy consumed in communications.

4.1 Energy Consumption

Fig. 2 shows the generation of 1 to 10 clusters in a network topology with 100 sensor nodes for each clustering algorithm. It also shows the results of energy consumption measurement during 10 rounds based on the number of each cluster generated.

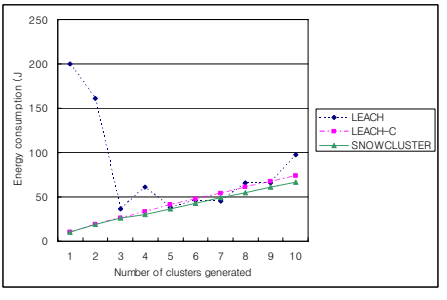


Fig. 2. Energy consumption during 10 rounds

In the case of LEACH clustering algorithm, until the number of clusters generated is 2, it shows significantly higher energy consumption compared to the other clustering algorithms, but after generation of more than 3, it is stabilized showing gradual increase. LEACH-C Clustering Algorithm shows progressive increase in energy consumption from round 1 to round 10. Similar to LEACH-C, SNOWcluster algorithm also shows results of gradual increase, but its consumption rate is slightly less than that of LEACH-C.

Fig. 3 is the result showing the amount of energy that is consumed during transmission of management message to the sensor node from the base station after formation of three clusters in the network topology of 200 nodes.

In the case of LEACH, because it does not have the position information of the nodes, inefficient routing is being resulted, and as a result, significantly greater amount of energy is consumed in transmitting management messages.

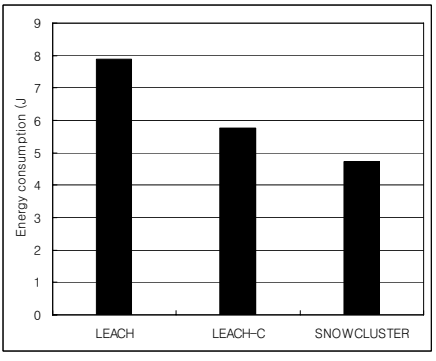


Fig. 3. Energy consumption for the management message

SNOWCLUSTER shows a result of decrease in the amount of energy consumption for management message transmission compared with that of LEACH-C. The reason of the result is that only a single region node in the SNOWCLUSTER plays the role of primary message transmission compared to the three cluster ones in the LEACH-C. Therefore, the cost of communication and energy consumption for SNOWCLUSTER is less than that of LEACH-C.

4.2 Network Lifetime

Fig. 4 shows results of changes in the network lifetime when 6 clusters are formed in network topologies of different number of sensor nodes of 50, 100, 150, and 200.

In the case of LEACH, almost the same length of lifetime is evidenced in topologies of 50 and 100 sensors, and network lifetime is the longest with 150 nodes. However, total network lifetime in a network formed by 200 nodes is shorter than that of 150 nodes. Such results display the fact that position of the node is not at all taken into account in the selection method of cluster head in the LEACH clustering algorithm and because of the lack of location information, energy consumption in forming the routing path between nodes is greater compared to those of other clustering methods.

SNOWCLUSTER shows a network lifetime that is 18∼20% greater than that of LEACH-C due to additional energy reduction effect resulting from the region node selection process. Therefore, it can be concluded that network lifetime can be prolonged by applying the SNOWCLUSTER algorithm in the sensor network.

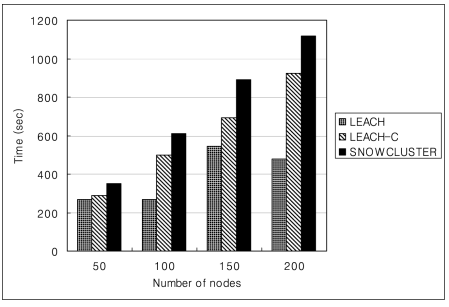


Fig. 4. Network Lifetime During Simulation Time

5 Implementation

5.1 Testbed Network

Our current work has focused on validating some of our basic ideas by implementing components of our architecture on Nano-24 [10] platform.

The Nano-24 uses Chipcon CC4220 RF for transmission and support 2.4 Ghz, Zigbee. The sensor node uses atmega 128L CPU with 32KBytes main memory and 512 Kbytes flash memory. The Nano-24 also supports Qplus-N sensor network development environment that ETRI (Electronics and Telecommunications Research Institute) developed. We organized a testbed network which was composed 10 Nano-24 nodes. Each node contains SNOWMAN's PE to support policy-based management as shown in Fig. 5. In this testbed, all sensor nodes are configured on hierarchical clustering architecture according to the SNOWCLUSTER.

5.2 Snowman

The PM and the PA of SNOWMAN architecture are implemented on Windows XP systems using pure JAVA. The PE is implemented on TinyOS in the Nano-24 nodes using gcc.

Fig. 5 shows the input forms for policy information on the PM. We use the XML technologies to define and handle global policies. There are several advantages of using XML in representing global policies [11]. Because XML offers many useful parsers and validators, the efforts needed for developing a policy-based management system can be reduced. To define XML policies, we customized and used the Scott's XML Editor [12]. The defined policies are stored locally in the policy storage of the PM and are stored remotely in the policy storage of the PA. PM communicates with PA via simple ftp for policy transmissions. To policy distribution to sensor nodes, we also design and implement TinyCOPS-PR that is simplified suitably for wireless sensor networks.

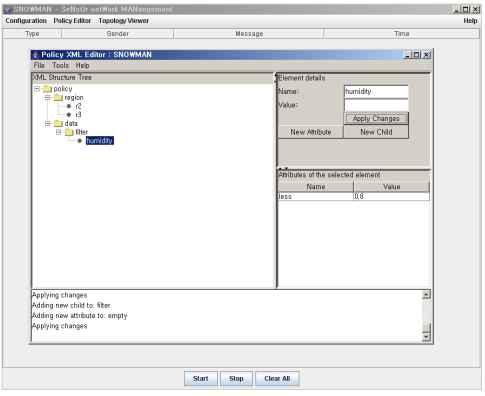


Fig. 5. Snapshot of SNOWMAN Policy Manager (PM)

6 Conclusion

In this paper, we proposed a policy-based management framework for self-managed WSNs called SNOWMAN. This paper also presented SNOWCLUSTER clustering algorithm. The SNOWMAN framework integrated the policy-based management paradigm and hierarchical cluster-based management architecture. SNOWMAN can provide administrators with a solution to simplify and automate the management of WSNs using PBNM paradigm. SNOWMAN can also reduce the costs of managing sensor nodes and of the communication among them using hierarchical clustering architecture. SNOWCLUSTER algorithm can be said to be more efficient in the aspect of network management and energy consumption than other existing sensor network clustering algorithms.

References

1. Si-Ho Cha, et al., A Self-Management Framework for Wireless Sensor Networks, LNCS 3842, January 2006.
2. Kaustubh S. Phanse, Luiz A. DaSilva, Extending Policy-Based Management to Wireless Ad Hoc Networks, 2003 IREAN Research Workshop, April 2003.
3. R. Yavatkar, D. Pendarakis, R. Guerin, A Framework for Policy-based Admission COntrol, IETF RFC 2753, January 2000.
4. Linnyer B. Ruiz, Fabircio A. Silva, Thais R. M. Braga, José M. Nogueira, Antonio A. F. Loureiro, On Impact of Management in Wireless Sensors Networks, IEEE/IFIP NOMS 2004, Volume 1, 19-23 April 2004.
5. M. Younis, P. Munshi, Architecture for Efficient Monitoring and Management of Sensor Networks, IFIP/IEEE E2EMON, September 2003.
6. Chien-An Lee et al., Intelligent Self-Organization Management Mechanism for Wireless Sensor Networks, http://www.ndhu.edu.tw/ rdoffice/exchange/CYC-paper.pdf.
7. M. J. Handy, M. Haase, D. Timmermann, Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection, 2002 IEEE.

8. Wendi B. Heinzelman, Anantha P. Chandrakasan, Hari Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, 2002 IEEE.
9. K. Chen et al., COPS usage for Policy Provisioning (COPS-PR), IETF RFC 3084, March 2001.
10. Nano-24: Sensor Network, Octacomm, Inc., http://www.octacomm.net
11. Si-Ho Cha, et al., Policy-based Differentiated QoS Provisioning for DiffServ Enabled IP Networks, LNCS 3909, August 2004.
12. Scott Hurring, XML Editor, http://hurring.com/code/java/xmleditor/.

A Proposal of Large-Scale Traffic Monitoring System Using Flow Concentrators

Atsushi Kobayashi[1], Daisuke Matsubara[2], Shingo Kimura[3], Motoyuki Saitou[3], Yutaka Hirokawa[1], Hitoaki Sakamoto[1], Keisuke Ishibashi[1], and Kimihiro Yamamoto[1]

[1] NTT Information Sharing Platform Laboratories,
3-9-11 Midori-cho, Musashino, Tokyo, 180-8585 Japan
[2]Hitachi, Ltd., Central Research Laboratory,
1-280 Higashi-Koigakubo, Kokubunji, Tokyo, 185-8601 Japan
[3]NTT Advanced Technology Corporation,
3-9-11 Midori-cho, Musashino, Tokyo, 180-8585 Japan
[1]akoba@nttv6.net, {hirokawa.yutaka, sakamoto.hitoaki, ishibashi.keisuke, yamamoto.kimihiro}@lab.ntt.co.jp, [2]d-matuba@crl.hitachi.co.jp, [3]saitou@nttv6.net, shingo.kimura@ntt-at.co.jp

Abstract. In a large-scale backbone networks, the traffic monitoring system needs to receive a large volume of flow records, so if a single central collecting process is used, it might not be able to process all flow records. In this paper, we propose a method that achieves better scalability by using flow concentrators, which aggregate and distribute flow records. A flow concentrator is located between one or more routers and traffic collectors. The connection methods enable the number of flow concentrators to be adjusted to suit the size of a given network. We propose a reference model for a flow concentrator that is valuable for large-scale networks. We also evaluate the effectiveness of a flow concentrator developed based on these models and the effectiveness of the methods.

Keywords: NetFlow, sFlow, IPFIX, Flow concentrator.

1 Introduction

Recently, there has been growing interest in flow-based traffic measurement methods, such as sFlow [1] and NetFlow [2], because these methods are useful for anomaly-detection and traffic engineering. These methods are based on traffic information exporting from routers.

The other sides, the traffic volumes handled by ordinary service provider networks have increased year-by-year. In Japan, the number of broadband users increased significantly and broadband traffic consumes 230Gbps. In next five years, traffic volume passing through the point of presence of major service providers is likely to reach 100Gbps. In such a traffic volume, we need to reduce traffic records to reduce the burden on the traffic monitoring system and handle properly them. Therefore, generally sampling and aggregation method are applied in router. But, these flow records reduced by sampling until required accuracy of anomaly detection are still too

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 53 – 62, 2006.
© Springer-Verlag Berlin Heidelberg 2006

large for monitoring system to handle them. In aggregation method, generally flow records are aggregated based on network prefix or AS number. But, we can not monitor detailed traffic information. In such established practices, there is the issue that we can't monitor detailed information over handling large traffic volume. Thus, we need a method that provides efficient aggregation and storage function, and need the architecture of allowing aggregation by multi-granularity.

We focus on flow concentrator that is one of IPFIX nodes in the [6]. There is a possibility that it achieves an aggregation by multi-granularity, however the detailed internal process model is not sufficiently described yet in other documents. We propose the model of flow concentrator that has aggregation, storage and distribution functions. The main contribution of this paper is the proposal of architecture for a traffic monitoring system using flow concentrators. Using flow concentrators enables achievement for large-scale traffic monitoring system.

The organization of this paper is as follows. An introduction to flow concentrators is given in Section 2. Section 3 proposes internal and external reference models of flow concentrators that achieve a large-scale traffic monitoring system. Section 4 discusses implementation issues of flow concentrators. Section 5 describes the flow concentrator called "FlowSquare" that we developed and presents evaluation results.

2 Introduction to Flow Concentrator

These days, several flow measurement solutions, such as sFlow and NetFlow, are widely used in networks. Recently, NetFlow version 9 [3] was developed and based on it, IPFIX [4] is being developed in IETF as a next-generation flow exporting protocol. In NetFlow v9 and IPFIX, any information elements can be added to a flow record flexibly without changing the protocol version. These protocols use a template that shows the format of flow information. In IPFIX, besides flow information, packet information can also be exported [5]. In IPFIX, a flow concentrator is a node that receives flow records, aggregates them, and exports the aggregated flow records. IPFIX nodes are shown in Fig. 1 [6].

A flow concentrator is a node whose basic process model contains collecting, metering and exporting processes. This is the basic process model. We define the flow concentrator that has several functions as follows to propose internal process model more detail.

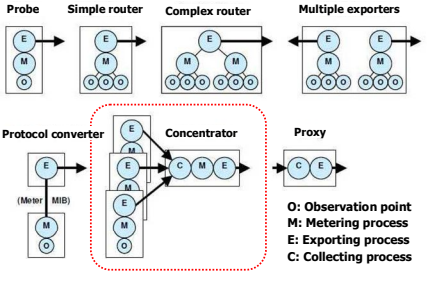


Fig. 1. Different types of IPFIX nodes

- It provides functions for balancing the aggregation load and storing aggregated data without requiring the receiving traffic collectors to have any special function.
- It has dual-role architecture for receiving and sending flow records. This architecture allows cascading concentrators, and the number of flow concentrators can be adjusted to suit the size of a given network.
- By storing flow records before aggregation, it enables us to refer to detailed traffic information in addition to summary information. This increases the scalability.
- External nodes can look up distributed flow records that have been stored in each flow concentrator.

Our reference model for a flow concentrator based on the above requirements is presented in the next section.

3 Reference Model for Flow Concentrator

Connection methods of routers, flow concentrators and traffic collectors are shown in the Fig. 2. A flow concentrator receives flow records by the IPFIX or NetFlow protocol and forwards flow records to another node. This dual-role architecture (acting as sender and receiver) enables the flexible connections. For example, it enables cascade connection methods.

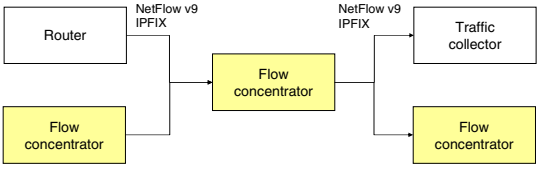


Fig. 2. External reference model of flow concentrators

To coordinate with flexible flow transport protocols like NetFlow v9 or IPFIX, the internal process model needs a flexible model. We propose a combination of several components as the internal process model. In Fig. 3, the flow concentrator contains six components: the collecting, selection, aggregation, reporting, exporting and storing processes. The collecting process receives flow records from routers. It also forwards received flow records to multiple selection processes or storing processes. The selection process has a filtering function and selects flow records that are matched under given conditions. The storing process has selects specified information elements by using storage rules and stores these flow records in a database. The aggregation process creates aggregated flow records in accordance with aggregation rules that are described in the aggregation draft [7]. The reporting process manages the reporting template, and the exporting process forwards flow records to the next nodes. Several processes are described in reference model document [8].

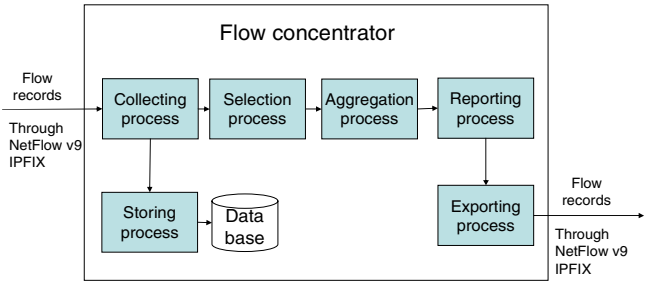


Fig. 3. Internal process model of flow concentrator

The combination of flow concentrators and several general-purpose traffic collectors gives the total system greater flexibility and scalability. Below, we show two examples of solutions using flow concentrators.

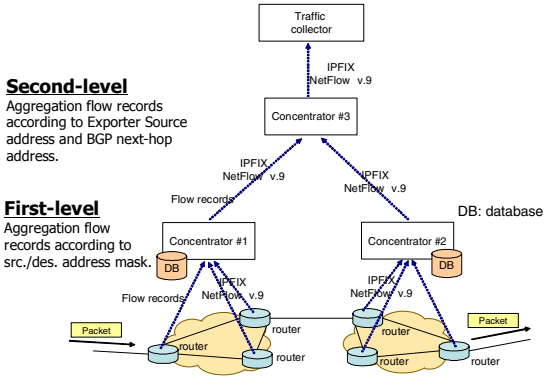


Fig. 4. Hierarchical structure of connections among flow concentrators

In Fig. 4, the hierarchical structure of connections is shown. In the first-level, a flow concentrator (e.g., #1) receives flow records from routers and creates aggregated low-level flow records. For example, the first level might be prefix mask aggregation. When another flow concentrator (#3) receives aggregated flow records, it aggregates them further. For example, the second step might be the aggregation of the BGP next-hop address and exporter address. After this, the traffic collector receives high-level aggregated flow records and stores them. This solution enables step-by-step aggregation without overloading any single node and the flow concentrators at each level store flow records of different granularity. For example, if the flow concentrators are located at each PoP that is across the world, this method allows reducing the burden of management network. It can make a large-scale traffic matrix and it can get specific flow records from the flow concentrator in case of need.

In Fig. 5, another method of connections among flow concentrators is shown. A flow concentrator distributes flow records according to the application role given by traffic collectors. For example, a traffic matrix measurement system needs the total traffic information in the network, so each concentrator sends all aggregated flow records to the traffic matrix measurement system. On the other hand, a concentrator sends only the important customer's traffic information to NBAD or IDS to achieve effective anomaly detection. For example, if several departments in ISP use same traffic information on different purpose, flow concentrator allows distributing them to several collectors.

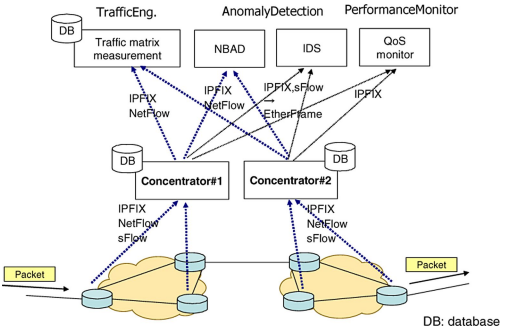


Fig. 5. Connections among flow concentrators

4 Implementation Issues

4.1 Instruction Templates

Each process performs functions based on instruction parameters that are set by an operator in advance. Some of them should be set by information elements and, defining instruction words enables the creation of the instruction templates, just like flow template of NetFlow v9 or IPFIX.

For example, the aggregation process has instruction words like "key", "keep", and "discard". This increases the flexibility, as in [7]. Similar to aggregation process, the storing process can have storing instruction templates. If the instruction word "store" is set in some information elements, these information elements should be stored in database. If "discard" is set as the instruction word, these information elements should not be stored. Examples of the storing and the aggregating instruction templates are shown in Fig. 6. The storing process stores only information elements that are labeled with the instruction "store" in the database. The aggregation process creates aggregated flow records that have common source/destination network addresses because the "key" instruction word is set for the "Source IP address", "Destination IP address", "Source address prefix", and "Destination address prefix" elements.

Additionally, we have proposed using those instruction templates as a management information base (MIB) [11]. External nodes need to be able to control several flow concentrators through remote access in order to control the whole system. This

enables collaboration between flow concentrators. To achieve this, we selected SNMP MIB rather than NETCONF [9] to reduce the burden of the node. The flow concentrator MIB has a collector MIB module and a concentrator MIB module. The collector MIB module can be used in an ordinary traffic collector. The concentrator MIB module has some instruction parameters that are managed by each process and it associates the instances of these processes, just like PSAMP-MIB [10].

Storing Instruction Template

ID	Information elements	instruction
01	Source IP address	"store"
02	Source port	"store"
03	Destination IP address	"store"
04	Destination port	"store"
05	Bytes	"store"
06	Packets	"store"
07	Protocol	"store"
08	INPUT IF INDEX	"discard"
09	OUTPUT IF INDEX	"discard"
20	Source address prefix	"discard"
21	Destination address prefix	"discard"

Aggregation Instruction Template

ID	Information elements	instruction
01	Source IP address	"key"
02	Source port	"discard"
03	Destination IP address	"key"
04	Destination port	"discard"
05	Bytes	"keep"
06	Packets	"keep"
07	Protocol	"discard"
08	INPUT IF INDEX	"discard"
09	OUTPUT IF INDEX	"discard"
20	Source address prefix	"key"
21	Destination address prefix	"key"

Fig. 6. Examples of the storing instruction template and aggregation instruction templates

4.2 Storing Solutions

In general, traffic collector uses an RDBMS that has an SQL query function. However, once flow records have been stored in the RDBMS, they are not changed within a certain storage interval. Thus, the RDBMS approach cannot be said to be the optimal solution for a large scale and high speed. Many recent open source tools for use as collectors have a flat file structure [12, 13]. This section discusses these solutions and considers which is suitable for a traffic collector that has several applications or for a flow concentrator that needs scalability.

Since PostgreSQL and MySQL have been widely used as RDBMSs and we focused on them in our evaluation. MySQL offers several engine types. We considered two of them MyISAM and HEAP. Of these, HEAP is fast because it is a memory-resident function.

As a flat file structure, we considered a method based on Perl storable which is useful for quick development and a flat file structure. The flat file structure obtained with Perl storable is outline in Fig. 7. With respect to directory structure, the root directory is set to the exporter address which is the router's address. The next directory is set to the date which means the year, month and day. The name of each file is its time of storage. This is useful when an operator is searching flow records for a specific time and deleting old flow records. With respect to storable structure, one flow record is converted into one hash table in order to make elements easy to search. The references of each hash table are added to the array table. Our test tool saves storable files at intervals of 3 or 5 minutes.

We examine time taken for storing, searching, and deleting each solution in the following environment. The searching time was included in the time for displaying

for searched flow records. Additionally, we evaluated the influence of the RDBMS when the number of accumulation records was increased.

- CPU: Intel(R) Pentium(R) 4 3.40GHz
- Memory :3574MB
- OS: FreeBSD 5.4
- Perl version 5.8.6
- MySQL version 4.1.16
- PostgreSQL version 8.0.7

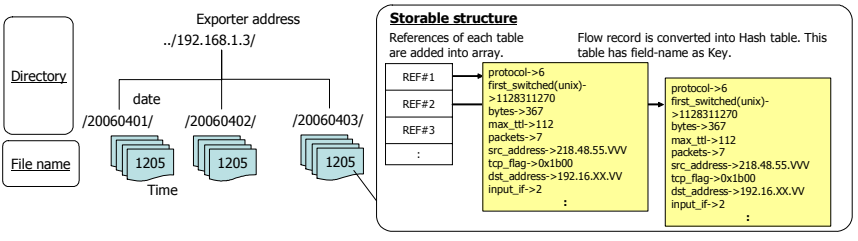


Fig. 7. Example of flat file structure obtained using Perl storable

Figure 8 shows the time required for storing and searching flows for each solution. The deleting time was about one second in all solutions and for all flow records. This indicates that the flat file structure using "storable" is preferable from the viewpoint of speed. Generally, the performance of an RDBMS is likely to become slower as the accumulation volume increases. Figure 9 shows the influence of RDBMS versus accumulation volume. In particular, the searching and deleting times clearly became slower as the number of accumulation records increased. In a large network, about 100 million records are accumulated per day. Therefore, it might be difficult to use an RDBMS.

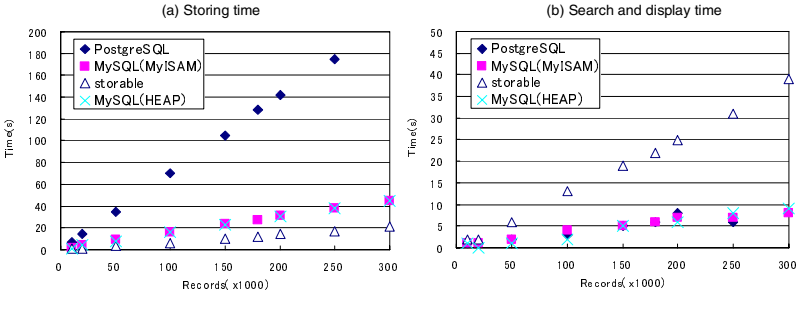


Fig. 8. Time required for storing (a) and searching (b). For storage, a flat file structure using Perl storable is faster than other methods. On the other hand, Perl storable is slow at searching and displaying, but this is less critical.

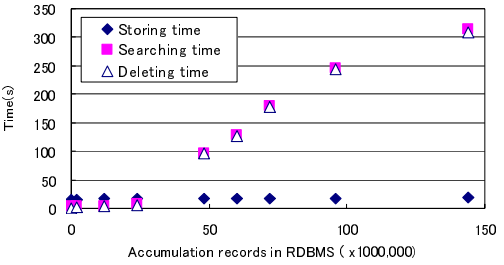


Fig. 9. Influence on MySQL performance as the number of accumulation records increases. We examined time required for storing, searching, and deleting for 100,000 records, after flow records had been added to the database until a specified number of accumulation records was reached. Even where PostgreSQL was applied, the tendency of the result was similar to the above.

Table 1 summarizes the results for the solutions, including streaming database, which has been proposed as a high-speed query function [14]. We can select these solutions case by case. For example, a flat file structure is preferable when the node needs scalability, just like a flow concentrator. On the other hand, RDBMS or a memory-resident database is preferable when the node needs multiple functions, just like a traffic analyzer.

Table 1. Summary of storing solutions

	Flat file structure (Perl storable)	RDBMS (MySQL, PostgreSQL)	Memory-resident database (MySQL HEAP)	Streaming database
Storing time	++ Several million records per minute	+ Several hundred thousand records per minute	+++ Several tens of millions of records per minutes	+++ Several tens of millions of records per minutes
Searching/ deleting time	++ Several million records per minutes	+ Becomes slower as accumulation volume increase.	+++ Several tens of millions of records per minutes	+++ Several tens of millions of records per minutes
Storage capacity	++ Dependent on storage disk capacity	++ Dependent on storage disk capacity	+ Dependent on memory capacity	+ Dependent on memory capacity
Query function	+ Only simple query	+++ Sophisticated queries like SQL	+++ Sophisticated queries like SQL	+++ Sophisticated queries like SQL
Speculation	Preferable for flow concentrator and simple traffic collector.	Preferable for traffic analyzer like NBAD.	Preferable for aggregation function that has a concentrator.	Needs further research.

5 Developed Flow Concentrator: FlowSquare

We developed "FlowSquare" as a flow concentrator based on the basic model. Besides having a storing function based on Perl storable, it can receive and send through NetFlow v9 and create aggregated flows of source/destination address pairs from a 5-tuple flow. We examined the CPU load of a flow concentrator and traffic analyzer to evaluate the effectiveness of the flow concentrator using the setup shown in Fig. 10. The results of examination are shown in Fig. 11 and Fig. 12.

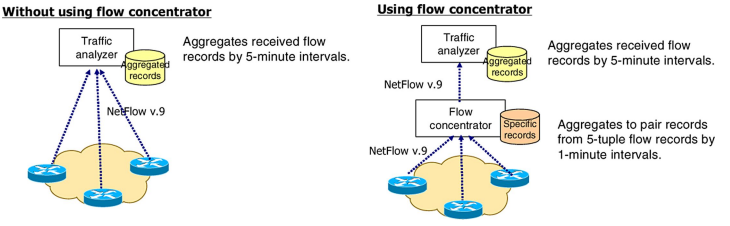


Fig. 10. Test environment. The traffic matrix measurements system was located in the traffic analyzer. We examined the CPU load on the analyzer and concentrator.

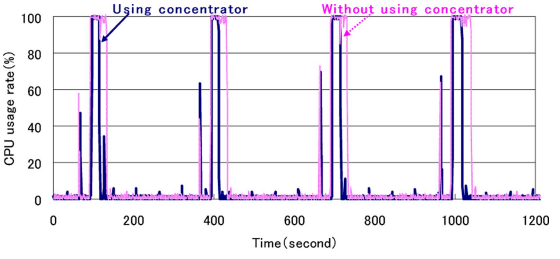


Fig. 11. Load on traffic analyzer in both cases. The load on the analyzer was reduced when a concentrator was used.

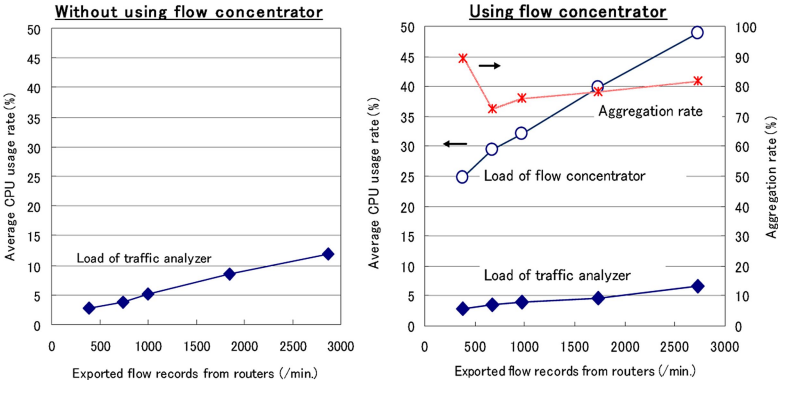


Fig. 12. CPU load of flow concentrator and traffic analyzer versus exporting speed of records. Also shown is the change in aggregation rate with record exporting speed.

In Fig.12, source/destination address pair aggregation by flow concentrator reduced CPU load of traffic analyzer about 5%. If the flow concentrator has several aggregation patterns, we can adjust them to according to the analyzer's performance.

For example, aggregation rates are likely to be expected about 30% in host aggregation and about 10% in prefix aggregation. In that case, whole system can become more flexibly.

In particular, the flow concentrator is useful for a top-level collector that needs all traffic information, like a traffic matrix measurement system. We can get more detailed flow records from flow concentrators and can search for more detailed information that is correlated to the whole traffic summary. This is, we can drill down.

6 Conclusion

In this paper, we proposed a reference model for a large scale traffic monitoring system using flow concentrators. These models include an external model as the solution and an internal process model to achieve greater scalability. In addition, we developed a flow concentrator that has basic functions and examined the effectiveness of using flow concentrators. In future works, we will develop multiple functions in flow concentrators to further increase the scalability and flexibility.

Acknowledgments

This study was supported by the Ministry of Internal Affairs and Communications of Japan.

References

1. P. Phaal, S. Panchen and N. McKee, “InMon Corporation’s sFlow: A Method for Monitoring Traffic in Switched and Routed Networks,” RFC3176.
2. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
3. B. Claise, “Cisco Systems NetFlow Services Export Version 9," RFC3954.
4. B. Claise, “IPFIX Protocol Specification,” draft-ietf-ipfix-protocol-16.txt(work in progress)
5. B. Claise, “Packet Sampling (PSAMP) Protocol Specifications,” draft-ietf-psamp-protocol-02.txt (work in progress).
6. J. Quittek, T. Zseby, B. Claise, and S. Zander, “Requirements for IP Flow Information Export(IPFIX),” RFC3917.
7. F. Dressler, C. Sommer, and G. Munz, "IPFIX Aggregation," draft-dressler-ipfix-aggregation-01.txt (work in progress).
8. A. Kobayashi, K. Ishibashi, K. Yamamoto and D. Matsubara, “The reference model of IPFIX concentrators,” draft-kobayashi-ipfix-concentrator-model-01.txt (work in progress).
9. R. Enns, “NETCONF Configuration Protocol,” draft-ietf-netconf-prot-12.txt (work in progress).
10. T. Dietz and B. Claise “Definitions of Managed Objects for Packet Sampling,” draft-ietf-psamp-mib-05.txt (work in progress).
11. A. Kobayashi, K. Ishibashi, K. Yamamoto and D. Matsubara, “Managed Objects of IPFIX concentrator,” draft-kobayashi-ipfix-concentrator-mib-01.txt (work in progress).
12. http://silktools.sourceforge.net/
13. http://nfdump.sourceforge.net/
14. D. Abadi et al., “Aurora: A Data Stream Management System,” VLDB Journal Vol.12, No.2, pp. 120-139, 2003.

Novel Traffic Measurement Methodology for High Precision Applications Awareness in Multi-gigabit Networks

Taesang Choi, Sangsik Yoon, Dongwon Kang, Sangwan Kim,
Joonkyung Lee, and Kyeongho Lee

BcN Division, ETRI, 161 Kajong-Dong, Yusong-Gu,
Daejeon 305-350 Republic of Korea
{choits, ssyoon, dwkang, wanni, leejk, kholee}@etri.re.kr

Abstract. Traffic measurement research has focused on various aspects ranging from simple packet-based monitoring to sophisticated flow-based measurement and analysis. Especially, most recent research has tried to address limitations of simple flow-based monitoring by utilizing payload inspection for applications signatures or by identifying target application group based on common traffic characteristics. However, due to highly dynamic nature of the development and the use of the Internet applications, individual simple remedy such as application signature inspection, dynamic port identification, or traffic characterization can't be sufficient to achieve high precision application-aware monitoring. This is true especially in multi-gigabit high-speed network environment. As the Internet has been evolving from free networks to high-quality business oriented ones, more sophisticated high precision application-aware traffic measurement is required which takes all the factors mentioned above into account. In this paper, we propose our novel traffic measurement methodology to meet such requirements. We considered scalability, cost-efficiency, and performance.

Keywords: hardware-based flow record generation, traffic measurement.

1 Introduction

In recent years, traffic measurement and analysis studies have received significant attention due to the requirements from various aspects such as SLA monitoring, P2P traffic monitoring, and detection of security anomalies, etc. However, technical challenges have been increased simultaneously because of the multi-gigabit link speeds, huge traffic volume to measure, and the dynamic characteristics of the current and newly emerging Internet applications.

High speed and volume traffic measurement requires improvements from both hardware and software. Even packet capture in one gigabit speed using a commercial NIC (Network Interface Card) causes significant packet losses when bursty traffic comes in. Dedicated special hardware has to be designed to meet such high speed and volume measurement requirements. Major capabilities that have to be considered in the hardware are high speed accurate packet processing including fragmented packet

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 63 – 72, 2006.
© Springer-Verlag Berlin Heidelberg 2006

assembly, filtering, various sampling methods (fixed, probabilistic, or flow sampling), wire-speed flow generation, and content inspection. None of the currently available hardware whether it is a card or a standalone device can meet the all the requirements mentioned above. Especially when it has to deal with very high speed links such as OC-48 or above. Most of them can meet the packet header processing requirement under normal traffic condition but show performance limitations in worst traffic condition, for example, in the case of DOS attack. PPS (Packet per Second) and FPS (Flow per Second) increases abruptly close to the theoretical upper limit.

For the diversity of the current and newly emerging Internet applications, the main difficulties come from highly dynamic nature of the development and the use of the current Internet applications. Traditionally, Internet traffic was dominated mostly by the client-server type of applications such as WWW, FTP, TELNET, etc. However, this characteristic has been changed significantly when new applications such as peer-to-peer, streaming, and network game applications were introduced. These applications use a range of port numbers or dynamically allocated ones for their sub-transactions (e.g., EDONKEY uses 4661, 4662, 4665, 6667 and RTSP streaming application allocates a port number dynamically for a stream data transfer). Some Internet applications intentionally use the same port number for malicious purposes, e.g., port number 80 for bypassing firewalls. This means that distinguishing flows based on a port number and other header properties is not safe and accurate enough.

Also since the Internet is asymmetric in nature, an application transaction consists of multiple sub-transactions, a series of Requests and Replies, which may follow different routing paths. Accurate flow identification for such a case requires distributed monitoring and correlation of sub-transactions which appeared in different paths. Another problem can be caused by packet fragmentation. IP packet fragments, but the first one, do not contain transport layer headers; no port numbers are given in such packets although they really are linked with a port. It is, however, reported that not a small portion of the Internet backbone traffic is transported in a fragmented state[1], and this tendency may deepen because more and more encapsulated services, such as IPv6 over IPv4, IPsec, etc., is getting popular.

In this paper, we propose a novel mechanism to cope with such challenges. It consists of hardware and software methodologies. For performance and scalability, packet inspection, filtering/sampling, traffic anomaly handling, and flow generation are all conducted in our novel hardware. Various traffic analysis including multi-path correlation and high precision applications recognition is the job of our software. We have incorporated the proposed mechanisms into our proof-of-concept system called Wise^{TrafView}. The paper is organized as follows. Section 2 examines related work. Section 3 and 4 describe our novel hardware and software methodology and overall system architecture to meet the above challenges. A proof-of-concept system and deployment experiences are explained in Section 5. Finally, section 6 concludes our research and development effort and lists possible future work.

2 Related Work

There have been many research and development efforts in the field of traffic measurement and analysis for the past decade. As a result, many tools were

introduced to meet various objectives such as traffic profiling, traffic engineering, attack/intrusion detection, QoS monitoring, and usage-based accounting. CAIDA's OCxmon[2], Tcpdump[3], Ethereal[4], and SPRINT's IPMon[5] can be used to capture full packets and analyze them off-line for the purpose of various traffic profiling. They are per-packet analysis systems. Cisco's Netflow[6], CAIDA's CoralReef[7], Flowscan[8], and NetraMet[2] are flow-based traffic analysis systems. They capture IP packet header information and compose them into flow records. Then various analysis tasks such as traffic profiling, usage-based accounting, and/or traffic engineering can be performed either on-line or off-line.

Per-packet analysis systems typically rely on dedicated special capture hardware. DAG card[9] is well-known for this purpose. Recently SCAMPI project[10] also introduced PCI-based capture cards for high speed and volume traffic measurement. Objectives of both cards are to capture the packets in real-time for the high speed links. It is claimed that both cards can perform measurement for upto OC-192. For the high precision applications recognition, wire-speed flow generation and payload inspection are essential functionality. However, these are major missing ones to them.

For the recognition of Internet applications, most solutions are targeted for P2P, streaming applications identification, or security anomaly detection. Cisco Systems' NBAR (Network-Based Application Recognition)[11] provides basic application recognition facilities embedded in their Internet Operating System (IOS) for the purpose of traffic control. Most intrusion detection systems (IDSes) are also equipped with basic application recognition modules which usually function on a signature matching-basis. For streaming and P2P application identification, Mmdump[12]and SM-MON[13] used payload inspection method. Many other methods for P2P traffic characterization [14,15,16] which depend on port number based matching were also published. Cisco's Service Control Engine (SCE) series[17] and Netintach's Packetlogic[18] provide more general purpose applications recognition solutions. These systems, however, exhibit shortcomings of performance, scalability, scope of coverage, and accuracy.

3 Methodology

In this section, we propose our novel methodology to fulfill the requirements in terms of hardware and software.

3.1 Hardware Methodology

In comparison with the existing traffic capture cards which were described in the section 2, we have designed ours with more strict objectives to meet the above requirements: packet captures without loss at upto multi-gigabit speed, lowest CPU resource utilization as possible, filtering, packet and flow sampling, deep packet inspection, anomaly detection, and flow generation.

Based on the design, we have developed a series of capture cards: DS-3, Fast Ethernet, OC-3 ATM/POS, OC-12 ATM/POS, and Giga Ethernet for the lower and medium speeds. We are also working on a metering system for 2.5 and 10 Gbps speeds. We have decided to develop a standalone system rather than a PCI-type card

due to various strong requirements we have set. It can not only capture the packets upto 10 Gbps speed without loss but also support wire-speed deep packet inspection for applications signature detection, flow sampling, anomaly traffic detection and special handling to relieve measurement burden, and, most importantly, flow record generation in the hardware. Our flow record format extends that of the netflow type records to support high precision applications analysis.

As mentioned above, packet capture cards that support upto 10 Gbps link speed have been developed. However, their capability is limited for packet header processing level. Any host system which installs the capture card can't support its wire-speed performance for the analysis work. Especially, when it comes to flow records generation, it represents very limited performance. It is mainly due to the fact that flow creation is usually done at the software level. In order to catch up with such a packet capturing speed, most of time consuming processes have to be pushed into the hardware level. Such attempt hasn't been tried by any researches before due to its high complexity and cost. It is especially challenging for us that we are trying to conduct hardwired flow creation with some extensions to achieve precise applications traffic accounting.

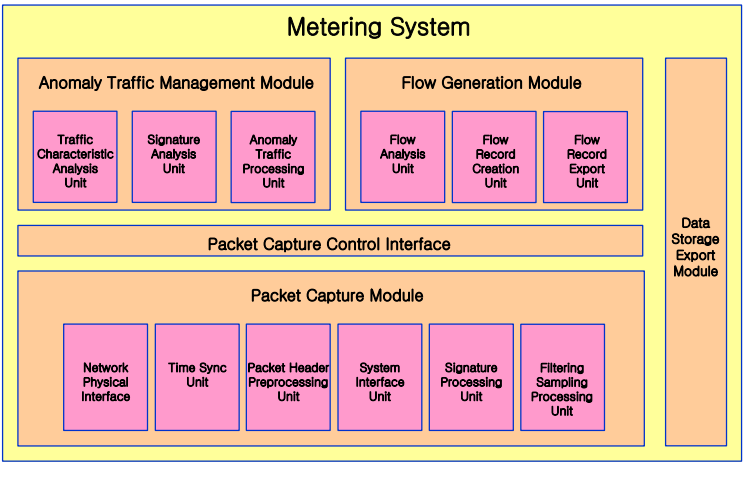


Fig. 1. Metering System Architecture

Fig.1 shows architecture of the proposed metering system to support multi-gigabit traffic capture. It consists of packet capture module, packet capture control interface, anomaly traffic management module, flow generation module, and data storage export module. For the consideration of performance, scalability, cost, and portability, we have decided to implement it as an ATCA(Advanced Telecom Computing Architecture)-based board[19]. Main processing engine is intel IXP2800 Network Processor (NP) and additional enhancements have been applied to meet our requirements.

Besides these most strict requirements, our system has other value-added functionalities. For scalability purpose, we added packet filtering and sampling

capabilities. It supports static, probabilistic, and flow sampling. And for precise application traffic recognition and anomalous traffic detection, we added content inspection capability in it. It supports up to 16 bytes payload searching.

3.2 Software Methodology

The novel mechanisms in our traffic measurement and analysis software consist of the following four items: general purpose Internet applications traffic classification, accurate application accounting, adaptability and expendability for newly emerging applications, and auto-detection mechanism of the new applications.

Most application monitoring systems currently available focus on a specific target such as P2P and/or streaming applications identification and security anomaly detection. Of course, there is an appropriate rationale for each effort. Our initial goal is the precise Internet application accounting for charging. Thus, we have put our efforts to come up with a general purpose Internet application identification methodology. Since we are using several novel methods which are described in details below to identify most of the popular Internet applications, the accuracy of application traffic accounting is much higher in comparison with most other currently available monitoring solutions which depend on port to application mapping. Since Internet applications lifecycle is very dynamic, it is very important to design the monitoring system to adapt such a characteristic. We added a run-time configuration language for such a purpose. When a new application appears in the Internet, we need to identify it promptly for its monitoring. Manual detection is very time-consuming and labor intensive work and thus requires automation. We are currently working on such automation and a preliminary result is explained below. Our software methodology consists mainly of a precise application identification method, extensible applications recognition language, and flow definition extension.

Precise Applications Classification Method

Our approach is unique in that we classify the Internet applications into the following four distinctive types. We determined the types by observing, test-utilizing, inspecting, generalizing, formulating, and classifying more than one hundred major Internet applications. We gathered flow and entire packet specimens from four major networks including a large-scale campus network, an enterprise network, a major Internet exchange point, and an ISP, and at least a week of collections were conducted for each test case.

An application is classified into not only per application level but also per sub-transaction level within an application. Each application comprises multiple sub-transactions such as request, reply, and data transfer. The objective to classify applications into such fine granular levels is to account precise traffic usage and to reduce the unknown traffic volume as much as possible. With this approach, we are aiming to identify applications with over 95% accuracy. Application recognition accuracy typically accounts for 20 ~ 40 % only if fixed-port recognition method is used.

- Type-FP (Fixed Port-based Recognition Type): Recognition is performed on the basis of a predefined port number to application mapping. Major well known services, such as WWW, FTP, SMTP, BGP, etc., and comparatively popular applications using

registered ports can be simply recognized by this method. There exists, however, a rather higher probability of misrecognition due to the current Internet applications characteristics as explained before.

- Type-PI (Payload Inspection-based Recognition Type): Recognition is performed on the basis of both port numbers and signatures, a.k.a. patterns, in the application PDU (Payload Data Unit). This method produces an effect when two or more equivalently influential applications share a registered port number. Any well known services or popular applications can also be identified by this method if higher level of correctness assurance is required.
- Type-DP (Dynamic Port-based Recognition Type): Recognition is performed on the basis of port numbers obtained by inspecting other flows' payloads. In the sense of payload inspection, this method is similar to type-PI; however, the difference is that, in this type, the sought pattern provides a referential hint to identify another flow (a type-DP flow or an induced flow) that may take place soon after. One of the common type-DP flow examples is a passive mode FTP.
- Type-RR (Reverse Reference-based Recognition Type): Recognition is performed on the basis of a referential note obtained by recognizing a type-PI flow on the other links. We define a reverse flow: When there exists a flow, X, of which <src_addr, dst_addr, src_port, dst_port, protocol> is <a, b, x, y, p>, if another flow, Y, is specified by (b, a, y, x, p), then Y is a reverse flow of X. The purpose of the type-RR method is, thus, to recognize reverse flows of type-PI flows. In most cases, type-RR flows are control flows of legitimate TCP connections whose constituting packets contain only IP and TCP headers or flows whose constituting packets do not contain any distinctive patterns in the payload

Extensible Application Recognition Language

In the section above, we described how to categorize the Internet applications into a set of distinctive types. Another important issue is then how to actually capture and analyze the contents of packets based on the above classification criteria. Considering highly dynamic nature of the development and the use of Internet applications, an adaptive and extensible content filter is essential component.

For this purpose, we propose an application recognition policy language, Application Recognition Configuration Language (ARCL), which succinctly describes the way to capture and analyze the contents of packets. The language is simple and effective in that the system can be swiftly reconfigured to detect and recognize unprecedented or modified applications without the developer's writing and distributing extension modules for processing the newer applications, which usually results in fairly long period of shrunk recognition coverage even after detecting and analyzing the new applications.

Specifically, an ARCL specifies the following features:

- The classes of applications into which each flow is categorized
- The classes of subgroups of applications into which each packet of a flow is categorized
- The methods and their parameters required to map flows to applications
- The methods and their parameters required to map packets to subgroups

The basic hierarchy of ARCL semantics and syntax possesses three levels. The highest category is an application (application), the next is a representative port

(port_rep_name), and the last is a subgroup (decision_group). There is a one-to-many relationship between a higher and a lower category. The basic concept can be explained by an example. Although most WWW services are provided through port 80, there are still many specific web services that are provided via other ports, such as 8080. All these representative ports pertain to the application "WWW" and have their own distinctive names; port_rep_names – "HTTP" for 80, "HTTP_ALT" for 8080, etc. Although a port_rep_name is tightly coupled with a port, the relationship between them is not one-to-one, nor fixed; a single port can be used by a number of different applications under different representative names. Packets in a flow can further be classified into a number of subgroups. For example, an HTTP flow subdivides into "HTTP_REQ" packets, "HTTP_REP" packets, ACK packets of HTTP_REP, etc. Each of these elementary subgroups constituting the entire context of a flow is a decision_group. The attributes and subsidiary categories of a higher category are specified in the corresponding clause whose scope is delimited by a pair of braces.

Flow Definition Extension and Application Detection Automation for Content-Aware Precise Application Analysis

Once each packet is examined and classified into the right type, captured information has to be aggregated into a flow and packet records for an analysis process. Typically, flow-based monitoring system generates flow records only. Our system adds packet records and associated payloads which include signature information. Note that we are not capturing all packet contents but the ones with application signatures only. Analysis server uses this information for the detailed analysis of traffic usage accounting per sub-transaction flow and eventually an aggregated application flow. We decided such an extension because none of the existing flow definitions serve our purpose. These records play a crucial role for the accurate application recognition.

Automated new application detection requires complex processes. When a new application can be known in advance, it is relatively easy to find its signature. If not, then it is getting much harder. In our system, unknown applications flow records are collected separately from the known ones. Unknown applications port numbers can be identified first and extract the flow records related with the unknown port number. Protocol interaction and flow correlation processes are applied to find out unknown application's signature. Currently we are developing analysis tools to automate these complex processes.

4 System Architecture

Fig. 2 shows an overview of the proposed architecture. Metering system captures packets either by an electronic or optical signal splitter. Upon receiving raw packets the Metering system at first filters out non-IP packets. Then, by processing the remaining IP packets, it composes flows and generates, in hardware, our extended flow records which compactly describe the flow and packet information respectively. During the flow composition process, application signature inspection specified in the ARCL description is conducted. As a result, the generated flow records contain flow summary information and information on the packets which belong to the flow and

payload portion of a packet which is matched by the signature. Our extended flow records, then, are sent to the Collector. It aggregates all the flow records collected from the multiple metering systems and export them to the Analysis server in real-time or store them in the Storage server for near real-time processing. The Analysis server classifies applications based on our classification algorithm and stores the result statistics in the DB. GUI then accesses it for various presentations.

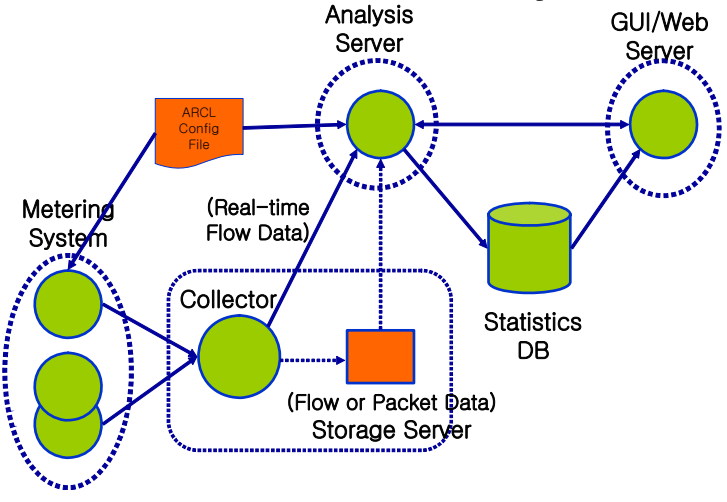


Fig. 2. Proposed System Architecture

The Collector is designed to be able to operate on a real-time basis because reducing the capture-off period as much as possible is essential in most operational cases. The operation of the Server, however, may not be constrained by time. The recognition operations can be conducted on a batch basis. Nevertheless, if the Metering systems and the Collectors are not too highly aggregated onto a Server and the Server possesses enough computing resources, the Server can also perform recognition tasks on a real-time basis. For very high-speed links, however, the Collectors and the Server may be severely overloaded, and accordingly, the way the Collectors and the Server operate must not be tightly coupled. Thus, our system architecture is designed to meet the mentioned requirements.

Our system can handle the same flow which goes through different measurement points. Each Metering system timestamps packets by GPS and the server which receives flow information from multiple Collectors can perform stateful classifications based on the accurate timestamps. Due to the space limitation of the paper, we focused our paper from the view point of novel methodology rather than system architecture and functional details.

5 Implementation and Experience

The proposed requirements and architecture is implemented as a proof-of-concept system, called Wise[*TrafView]. It has two versions: one for lower and medium speeds

upto 1 Gbps and the other for higher speed upto 10 Gbps. The former system has been installed and operated on a number of sites including a large scale campus network, a medium scale enterprise network (ETRINet), KIX - one of the biggest public Internet exchange (IX), and some other ISP's international 1 Gbps links in Korea. The latter system is under development and the prototype will be available in the third quarter of 2006 for the testing.

For the former, we have implemented intelligent packet capture (IPCAP) cards suited for our specific architecture and needs, they include DS-3, FastEthernet, OC-3/12 ATM/POS, and GigaEthernet Cards. They are PCI type-II based cards. Currently, we verified that cards upto 1Gbps can support wire speed packet capture without a single packet loss even for 64 byte IP packets arriving back-to-back.

In ETRINet, we monitored the most aggregated link which connects ETRINet to two major ISPs in Korea via two T3 (45 Mbps) links. The number of Internet users in ETRINet reaches around 2,500. Outgoing and incoming traffic is simultaneously fed into a Metering system and to a Server from a mirror-enabled Ethernet switch. The average profile of the consolidated traffic is 46.52 Mbps, 5.325 Kpps, and 174.1 fps, and the peak profile is 148.23 Mbps, 18.242 Kpps, and 1.359 Kfps. Two logical links (outgoing and incoming) are monitored by a single metering system. The Collector and the Server platform is composed of dual Pentium-IV CPUs, 2 GB of memory, and a 66MHz 64-bit PCI bus respectively.

By using our IPCAP GigaEthernet cards in ISP's international link, we have observed 0% packet loss on both incoming and outgoing links during the period of 6 weeks in March and April of 2006. Out-going link has around 200Mbps and incoming link has around 180Mpbs traffic utilization. Successful application recognition ratio is around 85% and we are trying to improve the unknown ratio. We have also tested POS OC-3 cards in another major Korean ISP's international link. This link was fully utilized and extensive system improvement and optimization work has been conducted. Thus, OC-3 card based monitoring system is recently commercialized. We are attempting various tests with OC-48 and OC-192 systems until the fourth quarter of 2006 and the performance analysis results can be incorporated in the final version of this paper.

6 Conclusion and Future Work

Due to highly dynamic nature of the development and the use of the current Internet applications, accurate application traffic usage accounting in the Internet requires a cleverly combined mechanism of per-packet payload inspection, flow-based analysis, correlation of associated sub-transaction flows, and wire-speed packet capturing performance. In this paper, we proposed the novel approach to meet such challenges.

In addition to designing architecture and proposing technical solutions, we have embodied them in a series of Wise^{*TrafView} systems. We are satisfied with the initial experiences with our systems.

We have tested our system with upto 1Gbps speed and are currently working for enhancing it to support much higher speeds such as OC-48 & OC-192. It is very

challenging that this new system shifts major functionality into hardware level. As far as we understand, such attempt hasn't been made by any researches yet. We are expecting to release our next version by the fourth quarter of 2006. Although we have focused on the usage-based accounting in this paper, it can be utilized in many other areas such as traffic profiling and security anomaly detection. These additional capabilities will be explored as our future work as well.

References

1. Colleen Shannon, David Moore, and k claffy: Characteristics of Fragmented IP Traffic on Internet Links. Proc. of ACM SIGCOMM Internet Measurement Workshop, San Francisco, USA, Nov. 2001
2. CIADA's OCxMon & NetTraMet. http://www.caida.org/tools/
3. TCPDUMP. http://sourceforge.net/projects/tcpdump/
4. Ethereal. http://www.ethereal.com/
5. Sprint ATL, "IP Monitoring Project," http://www.sprintlabs.com/Department/IP-Interworking/Monitor/
6. Http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm.
7. K. Keys, D. Moore, Y. Koga, E. Lagache, M. Tesch, and K. Claffy: The Architecture of CoralReef: An Internet Traffic Monitoring Software Suite. Proc. of Passive and Active Measurement Workshop 2001, Amsterdam, Netherlands, April 2001
8. D. Plonka, Flowscan: A network traffic flow reporting and visualization tool. In Proceedings of USENIX LISA, 2000
9. DAG cards, Endace Measurement Systems, http://www.endace.com
10. Jan Corppens, et. al., "SCAMPI – A Scaleable Monitoring Platform for the Internet", Technical Report, EU IST project, 2003
11. Cisco NBAR. http://www.cisco.com/warp/public/732/Tech/qos/nbar/
12. Jacobus van der Merwe, Ramon Caceres, Yang-hua Chu, and Cormac Sreenan "mmdump-A Tool for Monitoring Internet Multimedia Traffic," ACM Computer Communication Review, 30(4), October 2000.
13. Hun-Jeong Kang, Myung-Sup Kim and James Won-Ki Hong, "A Method on Multimedia Service Traffic Monitoring and Analysis", Lecture Notes in Computer Science 2867, Edited by Marcus Brunner, Alexander Keller, 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2003), Heidelberg, Germany, October, 2003, pp. 93-105.
14. Subhabrata Sen and Jia Wang, "Analyzing peer-to-peer traffic across large networks", in Proceedings of the second ACM SIGCOMM Workshop on Internet Measurement Workshop, Nov. 2002.
15. Alexandre Gerber, Joseph Houle, Han Nguyen, Matthew Roughan, and Subhabrata Sen, "P2P The Gorilla in the Cable", National Cable & Telecommunications Association (NCTA) 2003 National Show, Chicago, IL, June 8-11, 2003.
16. Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki, "Deconstructing the KaZaA Network", 3rd IEEE Workshop on Internet Applications (WIAPP'03), June
17. http://www.cisco.com/en/US/products/ps6151/index.html
18. http:// www.netintact.com/
19. http:// www.picmg.org/newinitiative.stm

Rate-Based and Gap-Based Available Bandwidth Estimation Techniques in Cross-Traffic Context

Wayman Tan[1], Marat Zhanikeev[1], and Yoshiaki Tanaka[1,2]

[1] Global Information and Telecommunication Institute, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan
[2] Advanced Research Institute for Science and Engineering, Waseda University
17 Kikuicho, Shinjuku-ku, Tokyo, 162-0044 Japan
waymantan@akane.waseda.jp, maratishe@aoni.waseda.jp,
ytanaka@waseda.jp

Abstract. Recent years have seen an increasing interest in end-to-end available bandwidth estimation. A number of estimation techniques and tools have been developed during the last few years. All of them can be classified into two models: rate-based and gap-based, according to the underlying approaches. The difference in characteristics of both models is studied in this paper. *PathChirp* and *IGI* are selected to represent each model and they are evaluated on low speed paths which are typical in today's local access of the Internet. Finally a hybrid method adopting both rate-based and gap-based approaches is proposed. The hybrid method is compared with *pathChirp* and *IGI*. Simulation proves that the hybrid method yields more accurate results and reduces overhead traffic.

1 Introduction

An end-to-end network path consists of a sequence of store-and-forward links that transfer packets from the source host to the destination host at each end of the path. Two commonly used throughput-related metrics of a path are the *end-to-end capacity* and the *end-to-end available bandwidth*.

In an individual link, the *capacity* is defined as the maximum IP layer transmission rate at that link. Extending the definition to a network path, the *end-to-end capacity* of a path is the maximum IP layer rate that the path can transfer from the source to the destination. It is determined by the smallest link capacity in the path.

The *available bandwidth* of a link is the unused portion of its capacity. Since a link is either idle or transmitting a packet at full capacity at any given instant of time, the instantaneous utilization of the link can only be either 0 or 1. Thus reasonable definition of available bandwidth should be the average unused capacity over some time interval. The available bandwidth A over a time period $(t - \tau, t)$ is

$$A(t-\tau,t)=\frac{1}{\tau}\int_{t-\tau}^{t} C(1-u(x))dx\,, \tag{1}$$

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 73 – 81, 2006.
© Springer-Verlag Berlin Heidelberg 2006

where C is the capacity of the link and $u(x)$ is the instantaneous utilization of the link at time x. Similarly, the *end-to-end available bandwidth* is determined by the smallest link available bandwidth in the path over certain interval.

End-to-end available bandwidth is usually estimated with active probing techniques which send probe traffic from the source host to the destination host of the path. Active probing is free of privileged access requirement and feasible for the end users. Various active probing techniques have been proposed today. Generally, they can be classified into two models according to the underlying approaches:

The *rate-based model* (RM) uses the sending rate of the probe traffic at the sender (probing rate) to infer end-to-end available bandwidth. The RM techniques usually send probe packets from an initial low rate and increase the rate gradually. They search for *the turning point* at which the arrival rate of probe packets at the receiver strata to be lower than the probing rate. Such turning point is believed to mirror the end-to-end available bandwidth over the probing period. RM techniques include *pathChirp* [1], *Pathload* [2] and *PTR* [3].

The *gap-based model* (GM) compares the time gap of successive probe packets between the sender and the receiver to calculate the available bandwidth. Given two successive probe packets fall in the same queuing period at the single bottleneck of the path, and they are sent with a time gap Δ_{in} and reach the receiver with a time gap Δ_{out}, then Δ_{out} should consist of two segments: Δ_{in} and the transmission time of the cross-traffic between them. The GM techniques use the difference between Δ_{out} and Δ_{in} and end-to-end capacity to calculate the rate of cross-traffic and then end-to-end available bandwidth. The calculation of GM tools is usually based on a set of congested probe packets. *IGI* [3] and *Spruce* [4] are examples of tools using GM.

2 Comparison Between Rate-Based and Gap-Based Available Bandwidth Estimation Tools on Typical Internet Paths

2.1 Representative Tools of Two Models

In this section two representative tools of each class are selected to be compared. In the RM class *pathChirp* [1] is famous for its *chirp* structure – a series of exponentially spaced packets within a single probe train. The chirp structure covers a wide range of probing rate and thus works with high efficiency. The probing process of *pathChirp* is fast and with significantly light overhead traffic. *IGI* [3] is a tool in the GM class. *IGI* sends an even spaced train with carefully selected initial gap and increases the gap linearly for subsequent trains. *IGI* estimates available bandwidth when all the gaps of train at the sender equal to those at the receiver. *PathChirp* and *IGI* are selected to represent their class in our study.

2.2 Test Environment and Methodology

Some evaluation of the existing end-to-end available bandwidth estimation tools has been done on high speed links (with capacity of 1000 Mbps) environment [5].

However, high speed end-to-end path does not yet prevail in today's Internet; most paths are with a capacity around 10 Mbps, usually constrained by the edge of the network. For practical purpose, we evaluate and compare the tools on paths with capacity of 10 Mbps. The test is performed based on simulations with OPNET Modeler. Simulation environment is crucial for this research as we need complete control over cross-traffic load on network paths and the ability to monitor link utilization with high level precision and granularity, which is not possible in real network environment. The network paths under observation are from 5 to 7 hops long with a single bottleneck in the middle of the path whose capacity is 10 Mbps. Admittedly, this topology is primitive and some special artifacts such as multiple bottlenecks are beyond the scope of our consideration, but the simple topology is favorable for discovering the fundamental characteristics of the tools.

Most available bandwidth estimation tools work well in stable cross-traffic environment. When they face highly bursty cross-traffic, however, the performance is very different for each tool. Therefore the performance in bursty cross-traffic environment is critical for evaluation. Except the idle path and light traffic scenarios, most of our test scenarios simulate highly bursty traffic environment. The cross-traffic is generated in the form of various popular Internet applications such as HTTP web browsing, FTP file transferring, Email, video conferencing and so on.

Both *pathChirp* and *IGI* have tunable parameters. Generally the default values of those parameters are used in our test. The initial gap of *IGI* is exceptional, because some test scenarios with high utilization are performed and the default initial gap is too large. For pathChirp, the probe packet size is 1000 bytes, the exponential spread factor is 1.3 and the length of the chirp is 15; for *IGI*, the probe packet size is 500 bytes, the initial gap is 0.5 ms and the length of a train is 32. This setting remains unchanged for all the comparison scenarios. Two tools run subsequently on the same path with the same cross-traffic condition repeated by the simulator.

2.3 Comparison Results

The first test scenario is performed on an idle path. The results are shown in Fig. 1. Both tools give an estimate around 10 Mbps, which is the end-to-end capacity of the path. While the estimate of *IGI* is very close to 10 Mbps, it is interesting to see that the estimate of *pathChirp* goes up to 11 Mbps. When *pathChirp* is evaluated on high speed links [5] the same phenomenon is also seen. Please note that the actual available bandwidth (avail-bw for short in the figures) is not 10 Mbps in both *pathChirp* and *IGI* cases. It is because the probe traffic occupies a little bandwidth of the path.

In the second scenario, there is light cross-traffic with average utilization at about 15%. The results of both tools, which are also shown in Fig. 1, are exactly the same as those on an idle path. It could be understood, because 15% of 10 Mbps is not more than the probe traffic rate of both tools. It means the cross-traffic is extremely light. Actually in a path with capacity of 10 Mbps both tools can not sense cross-traffic when the average utilization is lower than 15%.

Fig. 2 shows the results when the average utilization is 30%. Due to the bursty cross-traffic, the instantaneous utilization is heavily fluctuating. The accuracy of both

tools drops in this scenario. However, the estimate results of both tools show different characteristics. The estimate of *pathChirp* exhibits highly dynamic feature which reflects the fluctuation of the cross-traffic. A number of estimate results correctly follow the change of cross-traffic although some go opposite direction. But *pathChirp* is most of time over-reacting: even it follows the change of cross-traffic but the estimate value is either too high when the available bandwidth increases, or too low when it decreases. Some estimate results are higher than 10 Mbps which is obviously over the border. On the other hand, *IGI* barely corresponds to the change tendency of the cross-traffic. The range of the estimate results is limited. In this scenario, they are almost always higher than actual available bandwidth and fall into the area between actual available bandwidth and end-to-end capacity.

There is more cross-traffic in the next scenario and the average utilization is 50%. Fig. 3 shows the results in this scenario. Again *pathChirp* shows its strong ability to follow the change of cross-traffic but is over-reacting. The performance of *IGI* is also unchanged.

The final scenario is with heavy cross-traffic and the average utilization jumps up to 70%. Fig.4 shows the results in this scenario. Unsurprisingly, the particular characteristics of both tools exist as the same. But, *pathChirp* gives less estimates lower than the actual available bandwidth compared with previous scenarios. And, there are occasionally extremely high estimate results up to 30 Mbps from *pathChirp*.

In the above test scenarios, *pathChirp* and *IGI* exhibit different characteristics: *pathChirp* is good at following changes of cross-traffic but often over-reacts, while *IGI* does not catch changes of cross-traffic accordingly but offers a stable estimate. The different characteristics probably result from the underlying approaches of two models. The RM tools perform estimation based on a single turning point. For *pathChirp,* the turning point is a single packet pair inside the chirp. Therefore, it is fast to discover the change of cross-traffic but easy to deviate from the actual value. The *GM* tools estimate available bandwidth based on a set of congested probe packets. The estimate is smoother but can not follow changes in cross-traffic promptly.

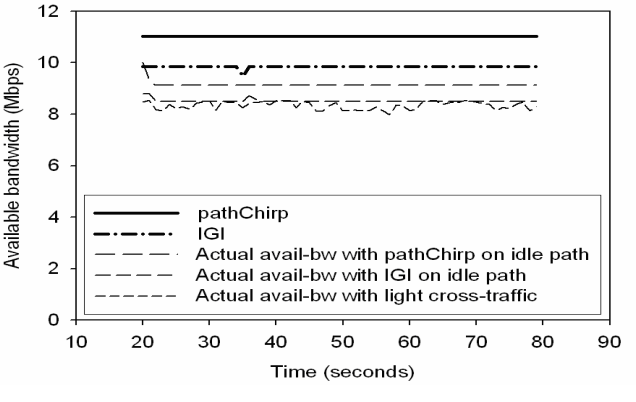


Fig. 1. Comparison between *pathChirp* and *IGI* without cross-traffic and with light cross-traffic

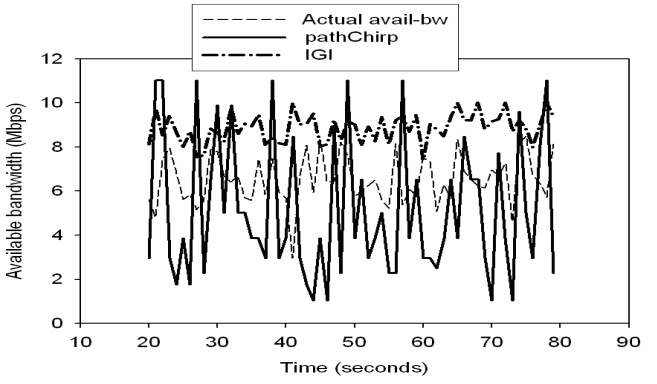


Fig. 2. Comparison between *pathChirp* and *IGI* with average utilization at 30%

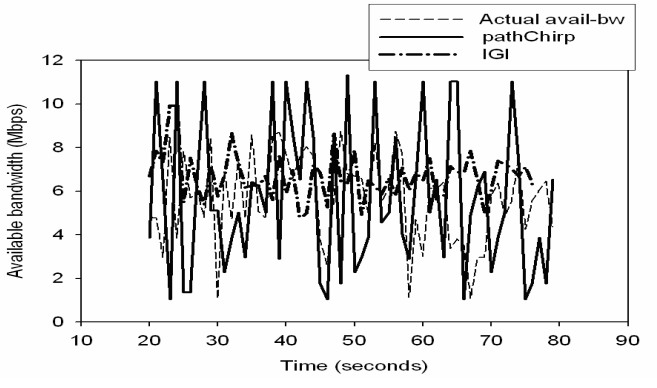


Fig. 3. Comparison between *pathChirp* and *IGI* with average utilization at 50%

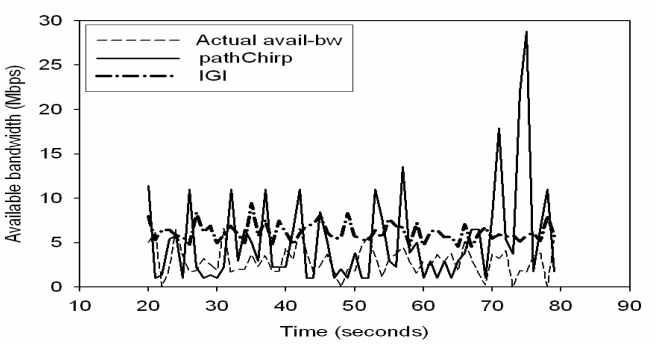


Fig. 4. Comparison between *pathChirp* and *IGI* with average utilization at 70%

3 Hybrid Method Adopting Rate-Based and Gap-Based Approaches

3.1 Introduction to Hybrid Method

In our experiment, we explore the different characteristics in performance of rate-based and gap-based available bandwidth estimation tools. In fact, we believe that their performance characteristics are complementary. Therefore, a tool that combines both models should have the potential to improve the accuracy of the estimate. A hybrid method adopting both *pathChirp* and *IGI* logic is proposed based on this assumption.

There are two phases in the method: the chirp processing and the *IGI* processing. The first phase is the chirp processing. It uses the same logic of *pathChirp* with certain enhancement: The initial rate of the chirp is set to a small portion of end-to-end capacity. It is set to cover a wide range of possible results of available bandwidth. The spread factor is no longer user specified. It is automatically set to ensure the highest probing rate of the chirp can be slightly higher than the end-to-end capacity. Self-induced congestion is guaranteed in this way. In this phase, the sender sends a chirp to the receiver. And after it receives the chirp, the receiver performs estimate with the logic of *pathChirp*.

The second phase is the *IGI* processing with the logic of *IGI*. The initial gap of the probe train is set according to the feedback of the chirp processing, instead of using default value. This is a significant enhancement in that it can effectively reduce the number of probe trains for the *IGI* processing. The initial gap is set based on the probing rate of the turning point in the chirp processing. Since the probing rate at the turning point is expected to reflect the available bandwidth, the initial gap based on this probing rate can converge faster. Similarly the sender sends probe trains and the receiver receives them and performs calculation.

After two probing phases are finished, two available bandwidth estimate results are generated. The *IGI* estimate then acts as an anchor for the chirp estimate to prevent the chirp estimate deviation. In addition, the chirp estimate is cut to be equal to end-to-end capacity when it is larger than it. A coefficient α is used to control the weight of the estimate of two phases in the final estimate. The final estimate E is given by

$$E = \alpha E_{\text{chirp}} + (1 - \alpha) E_{\text{igi}} , \tag{2}$$

where $0 \le \alpha \le 1$, E_{chirp} is the chirp estimate and E_{igi} is the *IGI* estimate.

3.2 Evaluation of Hybrid Method

We evaluate the hybrid method by comparing it with *pathChirp* and *IGI*. The comparison is performed in the same environment and with the same methodology as previous tests.

First, the hybrid method is compared with *pathChirp* and *IGI* when the average utilization of the path is 30%. The results are shown in Fig. 5. The coefficient α is 0.5 in this case. The hybrid method yields much closer results to the actual available bandwidth than the other two tools. It can not only follow changes of cross-traffic but

also effectively restrain the estimates from overreaction. In addition, the hybrid method reduces probe traffic in the second phase compared with the original *IGI*. The average number of trains of IGI process in the hybrid method is 1.7 while the number is 6.2 for original *IGI*.

Fig. 6 shows the comparison when the average utilization is 50%. The coefficient α is also 0.5. Again the hybrid method performs the best. The average number of IGI trains is 3.5 and that of original *IGI* is 9.1.

The final comparison is performed when the average utilization is 70%. The results are shown in Fig. 7. In this scenario, the coefficient α is 0.7. The hybrid method is still the best among the three. The average number of IGI trains is 3.7 and that of original *IGI* is 7.6.

We set the coefficient α with different values to see how it affects the final estimate. The error rate is used for verification. The error rate is the difference between the estimate and the actual available bandwidth value in proportion to the total capacity. A single error rate value is calculated from a number of estimate

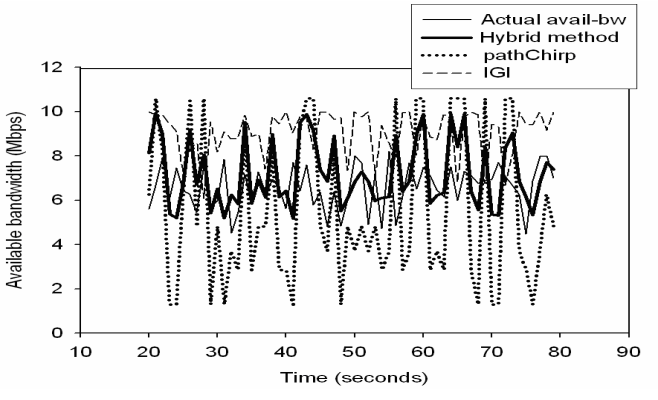


Fig. 5. Comparison among hybrid method, *pathChirp*, and *IGI* with average utilization at 30%

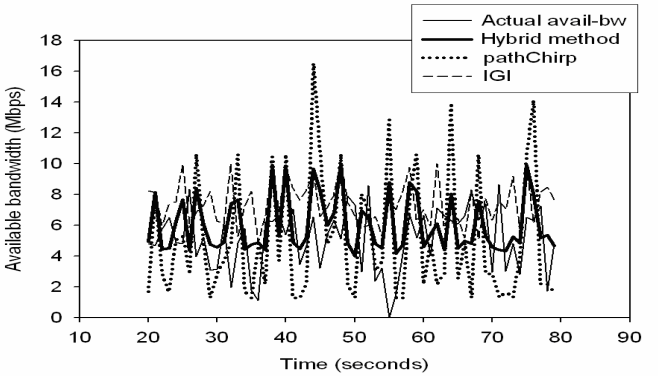


Fig. 6. Comparison among hybrid method, *pathChirp*, and *IGI* with average utilization at 50%

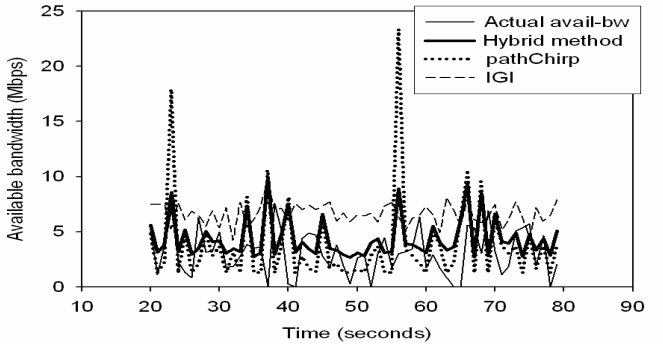


Fig. 7. Comparison among hybrid method, *pathChirp*, and *IGI* with average utilization at 70%

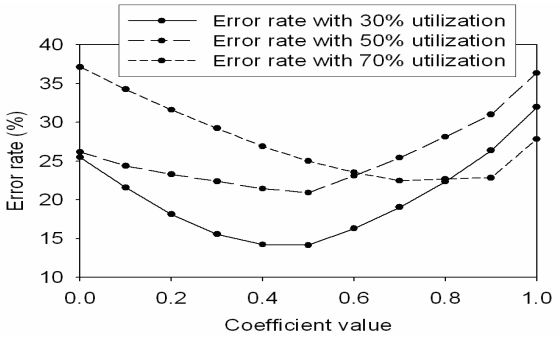


Fig. 8. Error rate vs. coefficient value with different utilization

samples of consecutive probes. The results are shown in Fig. 8. When the load is not heavy (e.g., the average utilization is under 50%), the value of 0.5 yields the best result. When the load is very heavy, the value of α should also increase because the *IGI* estimates are generally high and they should be restricted to a light-weight. Generally speaking, a network path with normal usage should not see very heavy utilization, so 0.5 is appropriate for the coefficient value. When heavy utilization happens, however, we can refer to the estimates of IGI as it is stable. For example, we can assume the utilization is heavy when the average estimate of IGI is no more than 6 Mbps in our test. This assumption is based on our empirical results.

4 Conclusion

In this paper, the differences in performance between *rate-based model* and *gap-based model* of end-to-end available bandwidth estimation techniques are studied. *PathChirp* and *IGI* are selected to represent each model. The evaluation is performed on low speed paths which are typical in today's Internet. A hybrid method adopting both

pathChirp and *IGI* logic is proposed. It takes advantage of the complementary properties of the two models Simulation shows that the proposed hybrid method significantly improves the estimation accuracy and reduces overhead traffic compared with original tools. Our future work is to evaluate the hybrid method in real world environment.

References

1. Ribeiro, V. J., Riedi, R. H., Baraniuk, R. G., Navratil, J., and Cottrell, L.: pathChirp: Efficient Available Bandwidth Estimation for Network Paths. Proc. Passive and Active Measurement Workshop, Tech. Report no. SLAC-PUB-9732 (2003)
2. Jain, M., and Dovrolis, C.: End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. IEEE/ACM Trans. Networking, Vol.11, No.4 (2003) 537-549
3. Hu, N. and Steenkiste, P.: Evaluation and Characterization of Available Bandwidth Probing Techniques. IEEE J. Select. Areas Commun., Vol.21, No.6 (2003) 879-894
4. Strauss, J., Katabi, D., and Kaashoek, F.: A Measurement Study of Available Bandwidth Estimation Tools. Proc. 3rd ACM SIGCOMM Conf. Internet Measurement (2003) 39-44
5. Shriram, A., Murray, M., Hyun, Y., Brownlee, N., Broido, A., Fomenkov, M., and Claffy, K.: Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links. Proc. Passive and Active Network Measurement, 6th International Workshop, Boston, MA, USA (2005)

Signature-Aware Traffic Monitoring with IPFIX*

Youngseok Lee, Seongho Shin, and Taeck-geun Kwon

Dept. of Computer Engineering, Chungnam National University,
220 Gungdong Yusonggu, Daejon, Korea, 305-764
{lee, shshin, tgkwon}@cnu.ac.kr

Abstract. Traffic monitoring is essential for accounting user traffic and detecting anomaly traffic such as Internet worms or P2P file sharing applications. Since typical Internet traffic monitoring tools use only TCP/UDP/IP header information, they cannot effectively classify diverse application traffic, because TCP or UDP port numbers could be used by different applications. Moreover, under the recent deployment of firewalls that permits only a few allowed port numbers, P2P or other non-well-known applications could use the well-known port numbers. Hence, a port-based traffic measurement scheme may not provide the correct traffic monitoring results. On the other hand, traffic monitoring has to report not only the general statistics of traffic usage but also anomaly traffic such as exploiting traffic, Internet worms, and P2P traffic. Particularly, the anomaly traffic can be more precisely identified when packet payloads are inspected to find signatures. Regardless of correct packet-level measurement, flow-level measurement is generally preferred because of easy deployment and low-cost operation. In this paper, therefore, we propose a signature-aware flow-level traffic monitoring method based on the IETF IPFIX standard for the next-generation routers, where the flow format of monitoring traffic can be dynamically defined so that signature information could be included. Our experimental results show that the signature-aware traffic monitoring scheme based on IPFIX performs better than the traditional port-based traffic monitoring method. That is, hidden anomaly traffic with the same port number has been revealed.

Keywords: signature, IPFIX, traffic measurement, flow, and security.

1 Introduction

Traffic monitoring is essential for accounting normal user traffic and detecting anomaly traffic such as Internet worms or P2P file-sharing applications. In general, simple packet- or byte-counting methods with SNMP have been widely used for easy and simple network administration. However, as applications become diverse and anomaly traffic appears quite often, more detailed classification of application traffic is necessary.

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). (IITA-2005-(C1090-0502-0020)).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 82 – 91, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Generally, traffic measurement at high-speed networks is challenging because of fast packet-processing requirement. Though packet-level measurement generates correct results, it is not easy to support high-speed line rates. In addition, standalone systems for packet-level traffic monitoring will be expensive for deployment and management in a large-scale network. Hence, Internet Service Providers (ISPs) generally prefer routers or switches that have already traffic monitoring functions to dedicated packet-level traffic monitoring systems. Recently, flow-level measurement methods at routers such as Cisco NetFlow [1] have become popular, because flow-level measurement could generate useful traffic statistics with a significantly small amount of measurement data. Routers with high-speed line cards such as 1Gbps are supported by Cisco sampled NetFlow. Thus, the standard [2] for traffic monitoring of routers has been proposed by IETF IP Flow Information eXport (IPFIX) WG, which defines the flexible and extensible template architecture that can be useful for various traffic monitoring applications. For example, IPv6 traffic monitoring, intrusion detection, and QoS measurement have been possible with routers due to the flexible template structure of IPFIX, which cannot be done with NetFlow v5.

Though flow-level traffic measurement is simple and easy for deployment, its measurement result may be incorrect, because only IP/TCP/UDP header fields are considered for traffic classification. Nowadays, due to firewalls that allow only well-known TCP/UDP port numbers, user and applications tend to change the blocked port numbers to the allowed well-known port numbers. In addition, recent P2P applications have begun to use dynamic port numbers instead of fixed port numbers. Therefore, port-based traffic classification may result in wrong traffic measurement results. On the other hand, when the payloads of IP packets are inspected to find the application-specific signatures, the possibility of correct traffic classification is increasing. Currently, most of intrusion detection systems (IDSes) or intrusion protection systems (IPSes) are employing packet inspection methods for investigating anomaly traffic patterns. However, IDSes and IPSes are focusing on only finding the anomaly traffic pattern as soon as possible and generating the alert messages.

In this paper, we aim at devising a flow-level traffic monitoring scheme that can utilize the signature information for the correct traffic measurement results while complying with the IPFIX standard. Thus, we propose a flow-level traffic monitoring method with extended IPFIX templates that can carry signatures for a flow. Our proposed method achieves the capability of correct traffic classification even at high-speed routers through examining the payload signatures as well as IP/TCP/UDP header fields.

The proposed scheme towards correct and IPFIX-compliant flow-level traffic monitoring has been verified with real packet traces in a campus network. From the experiments it was shown that anomaly traffic hiding itself with the well-known ports could be detected and classified. In addition, we proposed an IPFIX-compliant template that has been extended for carrying signature identification values.

The remaining paper is organized as follows. Section 2 describes the related work, and Section 3 explains the IPFIX-compliant signature-aware traffic measurement scheme. In Section4, we present the experimental results of the proposed method, and conclude this paper in Section 5.

2 Related Work

Typically, flow-level traffic measurement was done with Cisco NetFlow. FlowScan [3], that generates and visualizes traffic with NetFlow, uses port numbers for classifying applications. However, port-based traffic classification methods may be incorrect, because port numbers could be used by other applications. Although packet-level traffic measurement [4] could generate more precise results, it is expensive and difficult to deploy in a large-scale network.

In general, snort [5], which is a widely-used open IDS, can detect anomaly traffic such as Internet worms, viruses, or exploiting incidents including signatures. Thus, alert messages and logs are sent and recorded. However, the purpose of the IPS is to detect anomaly traffic.

Recently, a few content-aware traffic monitoring methods [6][7] have been proposed. In [6], signatures were used to classify traffic for accounting, and it was shown that traffic of well-known ports includes that of non registered applications. However, it does not support IPFIX. In [7], various traffic classification methods including packet inspection have been compared, and it was explained that unknown traffic could be correctly identified through searching signatures of the first packet, the first a few Kbytes, a few packets, or all the packets of the flow. However, these two studies use their own proprietary architectures for traffic measurement. In this paper, we propose a signature-aware traffic monitoring scheme that employs the IPFIX standard which could be used by next-generation routers.

3 A Proposed Signature-Aware Traffic Monitoring Method

In this section, we explain the proposed signature-aware traffic monitoring method.

3.1 Architecture

Figure 1 illustrates the key components of the signature-aware IPFIX traffic measurement architecture. Generally, the IPFIX device is embedded into routers or switches. However, a dedicated IPFIX device could be installed with capturing packets from the fiber tap or the mirrored port at a switch. The IPFIX collector gathers and analyzes IPFIX flows from multiple IPFIX devices through reliable transport protocols.

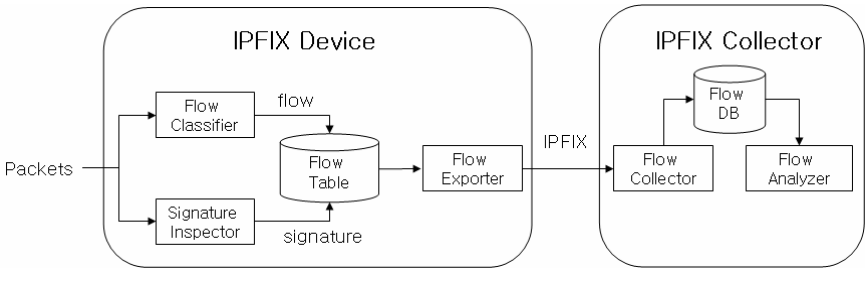


Fig. 1. Signature-aware IPFIX traffic measurement architecture

3.1.1 Flow Classifier

The flow classifier processes incoming packets with 5-tuples of IP/TCP/UDP header fields to find the corresponding flow entries stored at the flow table. If the flow entry corresponding to the incoming packet does not exist, a new flow entry will be created. Otherwise, attributes of the flow entry such as the number of packets, the number of bytes, the first/last flow update time, and etc. will be updated. A flow is defined by a sequence of packet streams sharing 5-tuples (source IP address, source port, destination IP address, destination port, protocol) of IP/TCP/UDP headers within a given timeout. A flow expiration timer is set to terminate a flow if a packet belonging to the same flow specification does not arrive within the timeout. Then, the expired flow entries will be exported to the flow collector. This flow idle timeout value can be configurable. For example, in our experiment, a flow idle timeout of 30 seconds was used as with Cisco routers. In addition to the flow idle timeout, another timer is required to finish and export long-lived flows residing at the flow table.

3.1.2 Signature Inspector

While packets are processed at the flow classifier, their payloads are simultaneously investigated by the signature inspector. The found signature will be recorded at the signature identification field of the corresponding flow entry. For this purpose, we defined a new IPFIX template with the signature identification field. A typical example of the signature inspector is snort that has signature identification values. In this paper, every single packet belonging to a flow is inspected for matching signatures. According to the given pattern-matching policy of inspecting packet payloads, it could be determined how many packets or bytes of a flow will be examined. Therefore, we can find signatures from the first K bytes, or the first K packets belonging to a flow. It is known that a single or the first few packets of a flow contain signatures of application protocols. For example, it is enough to examine a single packet for Internet worms consisting of a single packet, while the first packets should be investigated to find the patterns of P2P applications.

3.1.3 IPFIX-Compliant Flow Exporter

When flows are expired, the IPFIX-compliant flow exporter will send to the flow collector flow-exporting packets that contains flow information. Each flow entry includes data records according to the defined flow template. The flow template, which will be sent to the flow collector before the flow data are exported, explains how a flow is organized with several fields. A typical IPFIX-compliant flow data record consists of 5-tuple of IP/TCP/UDP header fields, the number of bytes, the number of packets, the flow start time, the flow end time, and the value of signature ID. In IPFIX, communication between the flow exporter and the flow collector is done through reliable transport protocols such as Stream Control Transport Protocol (SCTP) or TCP[1].

3.1.4 IPFIX-Compliant Flow Collector

The flow collector receives the template and data record for flows and saves the flows. The flow collector can communicate with multiple flow exporters and can

[1] Optionally, UDP may be used.

aggregate many flows into a simplified form of flows. Since a lot of flow data are continuously exported to the flow collector, a post-end database system is integrated with the flow collector for further analysis.

3.1.5 Flow Analyzer with Signatures as Well as Ports

Given the flow data record, the flow analyzer classifies flows with the signatures as well as typical port numbers. Thus, signature ID's are important when flows are classified. For example, Internet worms or viruses, P2P traffic, and other anomaly traffic that carry signatures are easily classified due to signature ID's regardless of port numbers. In addition, though either a few P2P applications are employing dynamic port hopping, or non-HTTP applications are using 80 port, they could be classified with their signatures.

3.2 IPFIX Templates for Carrying Signatures

Every IPFIX message consists of an IPFIX message header, a template set, and a data set (an option template set and option data set) as shown in Fig. 2. A template set defines how the data set is organized. A newly created template is sent through an IPFIX message consisting of interleaved template set and data set (option template set and option data set). After the template set has been delivered to the IPFIX collector, following IPFIX messages can be made up with only data sets. When UDP is used as the transport protocol, template records and option template records must be periodically sent.

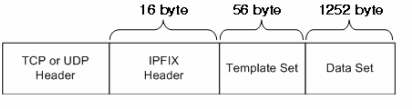
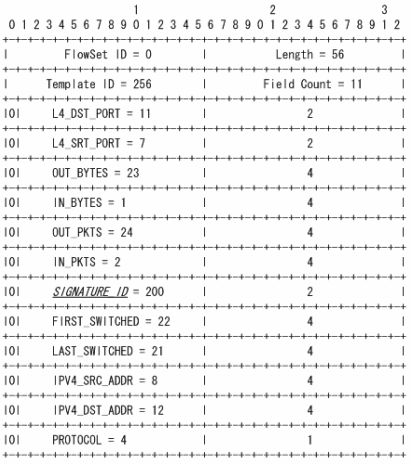


Fig. 2. IPFIX message

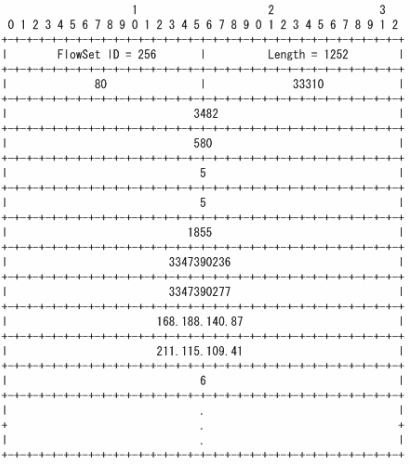
We defined a new flow template set including the signature ID field[2] as shown in Fig. 3-(a). The FlowSet ID of 0 means that this flow is the template. Basically, the flow defined by the template in Fig. 3-(a) delivers bytes, packets, flow start/end time, and signature ID for a flow of (src IP, dst IP, src port, dst port, protocol). Here, we use the signature ID values same with snort. Therefore, if the signature inspector finds a signature, it will record the signature ID at this field.

In Fig. 3-(b), the real example of the IPFIX data set which complies with the IPFIX template in Fig. 3-(a) is shown. The Template ID (=256) in Fig. 3-(a) and the FlowSet ID (=256) should be same if the flow data record is to be parsed according to the given template set. The web flow between 168.188.140.87 and 211.115.109.41 has 3,482 bytes, 5 packets, and the signature ID of 1855 which is related with the DDoS attack. Generally, in a single flow packet, more than one flow data set will be contained.

[2] The type of the "signature ID" is defined to 200 and the length of the "signature ID" is 2 bytes.



(a) IPFIX template set



(b) IPFIX flow data set

Fig. 3. IPFIX template and flow data message format including signature ID

4 Experiments

4.1 Prototype of a Signature-Aware Traffic Monitoring Tool

In order to evaluate the signature-aware traffic monitoring method, we implemented the prototype with snort and IPFIX-compliant flow generator, nProbe [8]. The prototype generates IPFIX flows according to the pre-defined flow template that includes signatures inspected by snort. For the IPFIX collector, we developed a real-time flow collector that can analyze the flows with signature ID.

4.2 Experimental Results

We verified the proposed signature-aware traffic monitoring method with packet traces in Table 1 collected at our campus network. This packet trace was captured at CNU as shown in Fig. 4 and it consists of mostly TCP traffic. Although many packet traces have been tested, only the representative set for two days is shown in this paper.

Overall, the prototype of the proposed traffic monitoring scheme has detected 0.6/0.8% flows with signatures for total inbound/outbound traffic. In the CNU campus network, since the recent negative firewall policy that opens only well-known port numbers has been employed, the anomaly traffic is not much reported in the experiments. Yet, our tool shows hidden anomaly traffic with signatures in Table 2. For example, “bad traffic with loopback addresses or UDP port 0” was found with signatures of 528 and 525. Possible exploiting traffic with signature 312 was observed. In outbound link, hidden P2P traffic called “Soribada” was seen with a user-defined signature 30004. In addition, possible DDoS attack traffic with signature 1855 was captured.

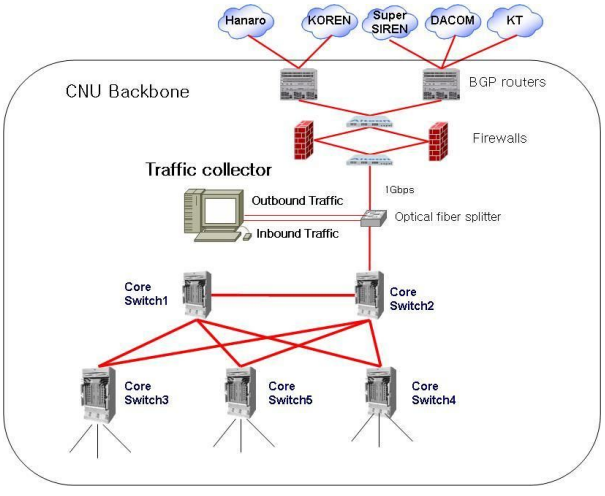


Fig. 4. Traffic measurement at Chungnam National University

Table 1. CNU campus packet trace in the experiments (2006.7.9 – 2006.7.10)

	Inbound	Outbound
Total bytes	3.2TB	2.4TB
Total packets	6,812,926,748	7,272,913,398
Total flows	65,130,555	80,017,160

Table 2. Top 10 signatures found in CNU network

Inbound		Outbound	
Signature ID	Number of flows	Signature ID	Number of flows
528	278,574	528	271,781
483	34,064	30004	255,711
525	19,338	1855	27,890
485	12,669	525	17,575
1419	9,454	1419	12,316
312	8,880	480	11,549
1417	8,871	2586	11,448
1201	4,532	312	8,486
1200	2,460	1417	8,373
486	2,162	2181	6,480

The detailed per-port statistics are shown in Table 3. In inbound traffic, various ICMP-based attack patterns have been found at port 0. Similarly, signatures are observed at well-known ports of 20, 22, 80, and 8080 as well as not-well-known ports of 2420, 2725, 3389, 4075, and 5063. In outbound traffic, one interesting port is 19101 which is used for web disk service of exchanging files.

Table 3. Top 10 port breakdown of traffic with signatures

Inbound			Outbound		
Destination Port	Total number of packets	% of packets with signatures	Destination Port	Total number of packets	% packets with signatures
0	198,190,150	2.9	80	1,446,312,138	20.0
80	180,441,016	2.6	19101	338,463,084	4.7
20	141,035,416	2.1	8080	329,413,874	4.5
8080	48,638,816	2.1	7132	290,554,498	4.0
2420	48,638,816	0.7	5090	273,500,608	3.8
2725	18,907,224	0.3	7778	182,935,034	2.5
5063	17,004,268	0.3	0	171,201,682	2.4
3389	16,867,212	0.2	23	154,077,628	2.1
4075	15,295,958	0.2	5004	140,164,098	1.9
22	14,619,240	0.2	6699	113,544,400	1.6

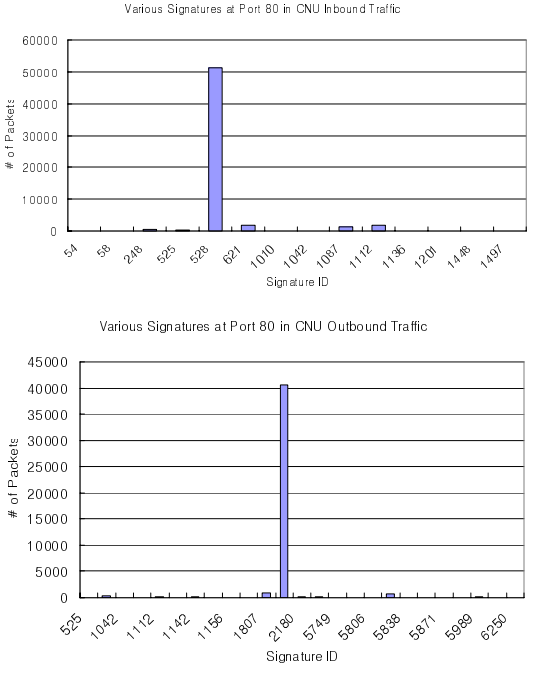


Fig. 5. Various signatures found at port 80

At the specific port number, the found signature information is widely distributed. For example as shown in Fig. 5, “BitTorrent” signature 2180 has been found in outbound link. In addition, at port 80, other signatures such as “bad traffic with loopback address (528)”, “web-misc whisker tab splice attack (1087)”, “spyware-put trackware (5837)”, and “DDoS attack (1855)”. From the experiments, it was shown that our signature-aware traffic monitoring method can illustrate the hidden P2P or anomaly traffic patterns.

Figure 6 is a snapshot of our tool [9] which can visualize signature-aware IPFIX flow data exported from routers. The traffic with signatures of 527 and 2586 has been shown. The signature ID of 527 is related with a DoS traffic attack with the same source and destination addresses. The signature ID of 2586 is the eDonkey traffic which has “E3” signature in the payload as follows.

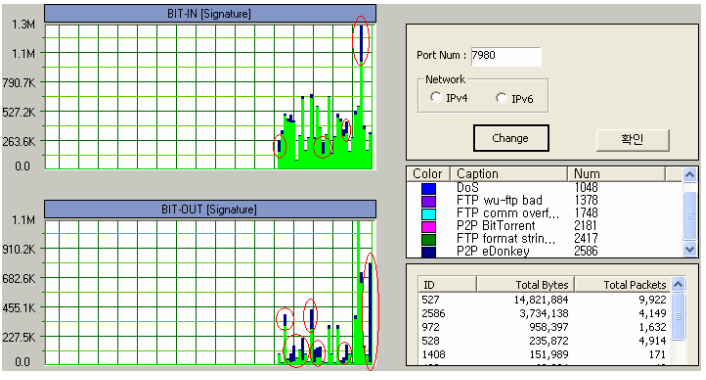


Fig. 6. A snapshot of visualizing the signature-aware IPFIX flows

5 Conclusion

In this paper, we proposed a signature-aware traffic monitoring method under the IETF IPFIX standard, and showed experimental results with the prototype. Our traffic monitoring scheme can reveal the hidden traffic patterns that are not shown under the port-based traffic monitoring tools. In order to be compliant with the IPFIX standard, we defined the signature field by extending the IPFIX template. While the traffic monitoring function proposed by this paper requires high performance for deep packet inspection and fast flow classification, it could be supported with network processor (NP) systems with ASIC or TCAM. In addition, since the proposed method uses the IPFIX standard, it could easily support IPv6 networks by changing IP address types. Although this paper has shown the first and realistic security-related application of IPFIX, the payload inspection algorithm is needed to be further studied for completeness and correctness. For instance, the false positive of the signature-based traffic classification method will be further studied in the future work.

References

[1] Cisco NetFlow, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/netflct/tech/napps_ipfix-charter.html
[2] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Requirements for IP Flow Information Export (IPFIX)," IETF RFC3917, Oct. 2004.
[3] D. Plonka, "FlowScan: A Network Traffic Flow Reporting and Visualization Tool," USENIX LISA, 2000.
[4] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-Level Traffic Measurements from the Sprint IP Backbone," IEEE Network, vol. 17 no. 6, pp. 6-16, Nov. 2003.
[5] M. Roesch, "Snort - Lightweight Intrusion Detection for Networks," USENIX LISA, 1999.
[6] T. Choi, C. Kim, S. Yoon, J. Park, B. Lee, H. Kim, H. Chung, and T. Jeong, "Content-aware Internet Application Traffic Measurement and Analysis," IEEE/IFIP Network Operations & Management Symposium, 2004.
[7] A. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," Passive and Active Measurement Workshop, April 2006.
[8] nProbe, http://www.ntop.org/
[9] WinIPFIX, http://networks.cnu.ac.kr/~winipfix/

Temporal Patterns and Properties in Multiple-Flow Interactions

Marat Zhanikeev[1] and Yoshiaki Tanaka[1,2]

[1] Global Information and Telecommunication Institute, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan
maratishe@aoni.waseda.jp
[2] Advanced Research Institute for Science and Engineering, Waseda University
17 Kikuicho, Shinjuku-ku, Tokyo, 162-0044 Japan

Abstract. It is widely recognized that today's Internet traffic is mostly carried by a relatively small number of elephant flows while mice flows constitute up to 80% of all active flows at any given moment in time. Although there are many research works that perform structural analysis of flows based on their size, rate, and lifespan, such analysis says very little about temporal properties of interactions among multiple flows originating from different applications. This paper focuses on temporal analysis of flows in attempt to grasp properties and patterns of flows that are related to application and user behaviour and can be captured only in the temporal view of traffic.

1 Introduction

The model of self-similarity is well established in traffic analysis literature. First parameterized models of aggregated traffic based of self-similarity can be found in [1] and [9], and were even considered for new queuing and routing disciplines [6]. The model was verified using LAN and high-speed WAN traffic in [7].

As the attention is mostly focused on flows at the tail of the distribution, many research works attempt to detect only large-volume traffic. Crovella et al. [8] developed the AEST tool that is based on self-similarity and finds a point in the distribution where heavy tail begins. The AEST tool was used by many researchers to extract large-volume flows [3] [4] [5].

First attempts to find correlation among flow metrics was made in [10], where correlation of size and rate of flows was sought. Zhang et al. in [10] found strong correlation between size and rate, as well as additional temporal characteristics, such as the fact that correlation grew stronger in larger intervals.

Practical results of large-volume traffic detection using heavy-tail model exposed the issue of temporal volatility of elephants [2]. The term volatility refers to the fact that elephant flows may not always appear at the top list which makes it difficult to provide steady identification of elephant flows. This artifact was studied in detail in [5] and a method called "latent heat" was introduced in order to allow certain freedom to elephant flows as long as temporal gaps were not too large.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 92–101, 2006.
© Springer-Verlag Berlin Heidelberg 2006

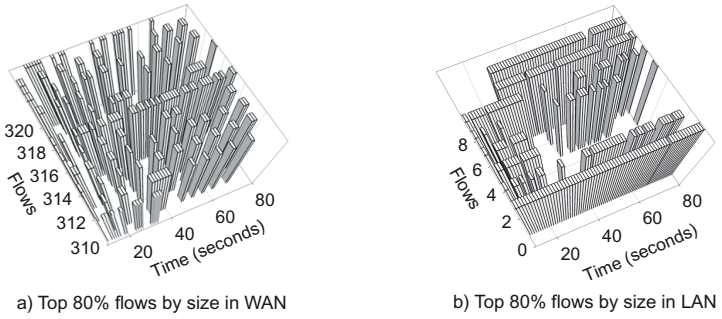


Fig. 1. Top flow membership volatility in LAN and WAN environments

In this paper a flow is attributed with a parameter that stands for the intensity of network interactions of an application or a user at the source. Hereinafter this parameter will be referred to as interactivity parameter. Applications that heavily rely on feedback from flow destinations would have low interactivity, while file transfers and otherwise generally non-stop transmissions creating one-time large bulks of data are considered to be more interactive. Naturally, human response would have lowest possible interactivity. The paper proves that elephant flow volatility can be explained with the introduction of interactivity parameter at the source. Additionally, the noval concept of interactivity allows to depart from traditional approach that considers only flows found in the distribution tail by giving a means to separate flows into groups based on principle flow metrics.

2 Heavy Tail with Temporal Interactivity

2.1 Problems with Heavy Tail in Practice

Self-similarity property states that most of traffic is confined within a small number of flows. Many research works use this property to separate elephant flows. This can be done using AEST method, which finds the point in the distribution where the tails starts. There is also another method called constant load, which uses a threshold to separate elephant flows. However, results from both methods exhibit acute volatility of elephant flows. To give an example, in Fig.1 we display top flows from LAN and WAN traffic extracted based on 80% constant load rule. Almost all the flows in the plots are elephant flows, but only a few are steadily appearing in the top flow list (list of flows with size above 80% of max size), while most are missing in many intermediate intervals. This problem is even more obvious in WAN traffic, which means that volatility increases in higher levels of aggregations. In this paper we attempt to find explanation for elephant flow volatility in the properties of traffic sources.

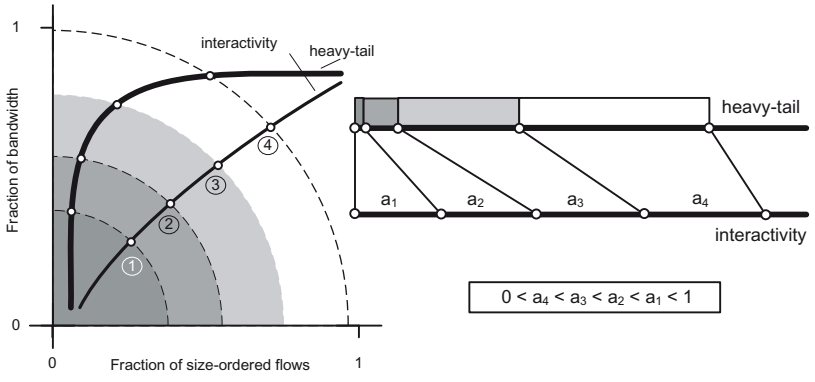


Fig. 2. Model of flow distributions based on heavy-tail and interactivity properties

2.2 Temporal Interactivity

The concept of interactivity can be introduced as follows. Let v denote the volume of traffic transmitted by application within interval T using only a part t of interval, i.e. $t < T$. In some cases it is possible for t to equal T, in which case gaps (periods of silence) would be defined only by interactivity parameter. Let α denote the level of interactivity at the source of the flow with values normalized to the fraction of 1, i.e. $0 < \alpha < 1$. Then, given the transmission rate of the link R, the volume transmitted by the flow within T can be expressed as :

$$v = \alpha t R. \tag{1}$$

As long as total load within the interval $V = \sum_{i=1}^{N} v_i < RT$, i.e. the network is not congested, flows are expected to compete for their share of traffic based on the value of α parameter. For automatic applications that do not require any feedback, α takes values close to 1, and low values stand for feedback-intensive applications or users. Feedback is a combined parameter that accounts not only for TCP ACK packets, but also for the time that applications may require to process data, get user response, etc. Parametric study of α is beyond the scope of this paper, but we perform the analysis of its physical properties and relation to self-similarity properties.

Physical properties of interactivity in relation to the well established heavy-tail behaviour are displayed in Fig.2 in form of bandwidth to number of flows distribution. Upper curve is the normal heavy-tail distribution, where most of traffic is confined within a few first flows ordered by size. Lower curve represents distribution as it would be without heavy-tail, i.e. only with α defining results of multiple-flow interactions. In contrast to conventional approach, we consider range 1 to consist of not elephants, but rather short flows that do not require any interactions with destination, that is have very high α. Those flows that are traditionally considered elephants, i.e. P2P software, file transfer, etc., are placed in range 2 with moderately high α. As α can vary depending on type of appli-

cations and even CPU of a particular host, range 2 is considerably wider than range 1. Ranges 3 and 4 are populated by different large classes of applications, such as web-browsing, mail, and others. Detailed study of classes is beyond the scope of this paper.

3 Analysis in LAN and WAN Traffic

To verify the proposed model based on interactivity parameter α, real traffic collected in LAN and WAN environments was analyzed. LAN traffic was collected in the backbone of Waseda University and WAN packet traces were obtained at a major WAN node in Japan. Naturally, LAN and WAN traces are delivered from two different levels of aggregation. Packet traces were processed into unidirectional IP-port pair flows with 30s timeout. Unidirectional flows are more prone to clean classification due to the lack of ACK packets that affect packet size distribution. 1 hour of LAN and 5 minutes of WAN traffic was used resulting in over 6000 flows in LAN and over 300000 flows in the case of WAN. At any given point of time 100 flows and 5000 flows in average were found in LAN and WAN data respectively.

From 5% to 10% of all flows in both traces are confirmed to be elephant flows in regard to data size and the traffic they carry is normally between 60% to 80% of all size. This complies to both the theoretical establishments of self-similarity and data analysis results of other practical research in the field.

3.1 Size, Rate and Lifespan Correlation

To analyze correlation between flow metrics we devised a 2-stage filter, where each stage uses a flow metric along with a threshold (fraction of the max value) that is used to filter flows. Flows passing the first stage are filtered through the second metric in a similar fashion, finally resulting in a number of flows that is plotted on the Z axis. Flows are always picked in descending order, i.e. the threshold is used to define the lowest value (low threshold results in less flows picked from the top). In this filter, the order of metrics used is important and results in different image of mass distribution. We specifically mark X and Y for stage 1 and stage 2 of the filter respectively. We include both combinations in a pair of metrics in each figure.

Fig.3 displays results of size/lifespan filters. WAN plots exhibit perfect conformity with heavy-tail properties and results are nearly mirrored. LAN data behave similarly with the only exception that due to the small number of flows tail of the distribution is not properly represented. However, the image of heavy-tail is preserved.

Fig.4 offers a different image of correlation between flow lifespan and rate. Lifespan used in the first stage still exhibits distribution fairly consistent with heavy-tail, but rate/lifespan filter offers a different picture in which one can visually separate a few distinct groupings. This artifact can be explained by the proposed interactivity parameter α, which does not follow heavy-tail or even

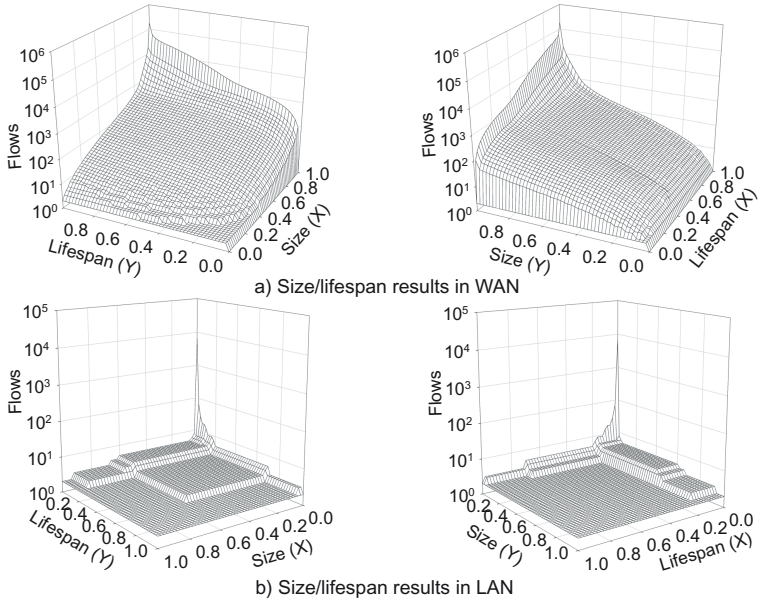


Fig. 3. Results of size/lifespan and lifespan/size filters

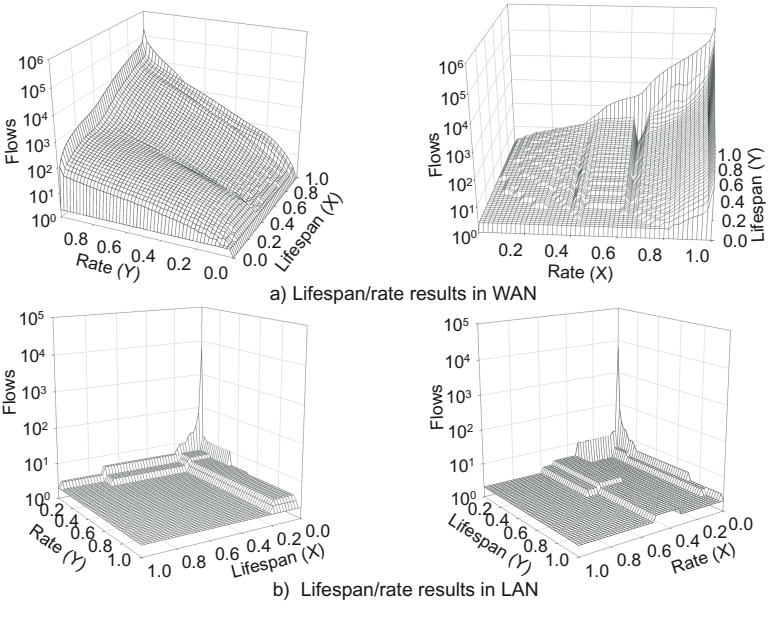


Fig. 4. Results of lifespan/rate and rate/lifespan filters

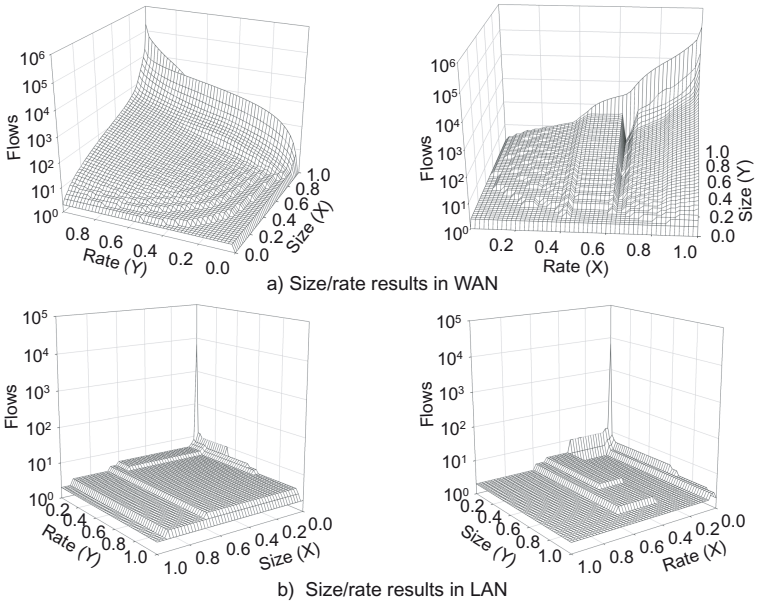


Fig. 5. Results of size/rate and rate/size filters

exponential distribution, but plays significant role in mass distribution of flows by rate. Similar conclusion was made in [10] that poses rate as a richer metric in terms of temporal and structural features than size. In the proposed model, he grouping between 0.4 and 0.6 of rate should contain only elephants.

Fig.5 displays similar behaviour for rate/size as it was in case of rate/lifespan. Similar irregularities in distribution can be found and are separated in a similar fashion. This is natural if to consider that lifespan/size correlation follows heavy-tail distribution, which is the reason that patterns found in rate/size filter are also reflected in rate/lifespan. LAN plots in both rate/size and rate/lifespan filters still lack solid evidence due to shortage of flows, but still exhibit similar groupings in gaps in rate/size and rate/lifespan parts.

3.2 Analysis of Top Flows by Rate and Size

Temporal analysis of multiple-flow interactions can offer more insight. As in temporal view the lifespan property has little meaning, we use only rate and size. The device we use for temporal analysis is very simple. For each interval we select a certain number (fixed or a fraction of total number) of top flows ordered by rate and those ordered by size. We use the estimate M to identify the ratio of flows from one list that match flows in the other, i.e. $M = n/min(N_s, N_r)$, where n is the number of flow matches, and N_s and N_r stand for total number of flows in size and rate ordered lists. Using a range of values for both metrics,

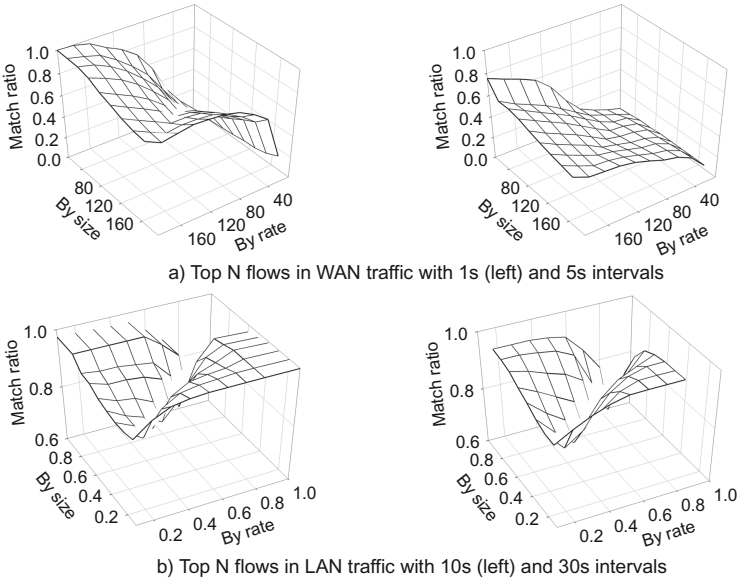


Fig. 6. Cumulative match ratio between top N flows ordered by size and rate

we obtain a 3D graph that tells us about cumulative similarity distribution in both ordered lists. Each point in plots is a mean over 10 samples taken from 10 intervals in a row. We use 2 different intervals to access scaling features.

Fig.6 contains matches obtained in comparing top lists with fixed number of compared flows. All plots in the figure have one common feature - diagonal area of low match ratio. This is due to the fact that on the diagonal line sizes of both lists are the same, and it results in less matches than in cases when one list of larger than the other. Besides, we are not interested in the diagonal line as much as we are interested in matches at extreme values of each metric. Thus, WAN traffic clearly results in poor matches with small number of top flows by rate regardless of the number of flows by size. This picture deteriorates even more with the increase of interval (right of Fig.6(a)). Areas of good matches in WAN case is the area of small number of top flows by size and large number of top flows by rate, which supports earlier discussed features of both distributions. This physically means that small number of top flows by size are grouped in the area of moderate rate values.

Behaviour of LAN in Fig.6 exhibits the same diagonal behaviour as in WAN data, but the deterioration process in the rate wing is much slower. However, at 30s intervals the rate wing of the plot is considerably smaller than the opposite size wing. One can expect this process to continue with the increase of interval. Also, due to the fact that at any given of time there were few flows (around 100 as opposed by 5000 in WAN) completing for shares in traffic, we used fractions of total number instead of a fixed number in X and Y axes.

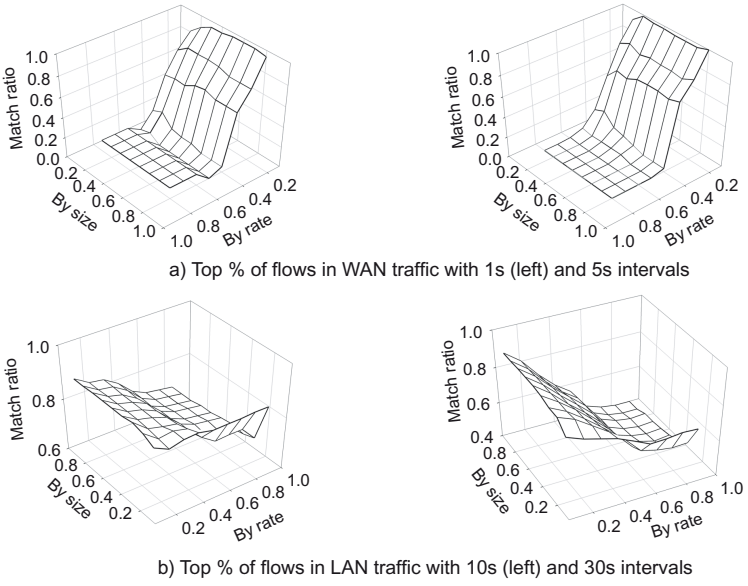


Fig. 7. Cumulative match ratio between threshold-selected top flows ordered by size and rate

Fig.7 is the result of matching performed using thresholds to create top lists instead of fixed number of flows. Although this figure generally supports our argument, it also offers new properties. The perfect parallel alignment along the size axis stands for the fact of very high variability of size, as for each threshold of rate almost the same number of matches can be found for almost all sizes. The smoothness of the plot also comes from the fact that temporal samples collected over 10 successive intervals vary too much to provide any visual difference. Since, as opposed to Fig.6, in this figure we have cases of total matches (area of low threshold of 0.2 for each metric, i.e. all flows from the ordered list). Naturally, all flows in that area match, thus providing the highest sample at (0.2,0.2). The LAN part of Fig.7 displays similar pattern, with the only additional feature of a spike at 0.2 on size axis parallel to rate, which says that when almost all flows by size are used, they can almost always be found with any length of rate lists.

To prove higher variability in size than in rate, we performed localized matching analysis in Fig.8. As opposed to previous cumulative analysis, this figure can pinpoint the area where match ratios are higher. Thresholds are used again to create top lists. We use the window of 0.1 to create local area and the step of 0.05 to provide smooth and detailed map.

Fig.8 contains similar features for both LAN and WAN traffic, but they are expressed differently in each environment. WAN results clearly convey the message that small flows by size, i.e. small number of bytes transmitted within the interval, are often due to low transmission rate of the flow (low interactivity parameter α). This feature grows stronger in longer analysis interval. In addition,

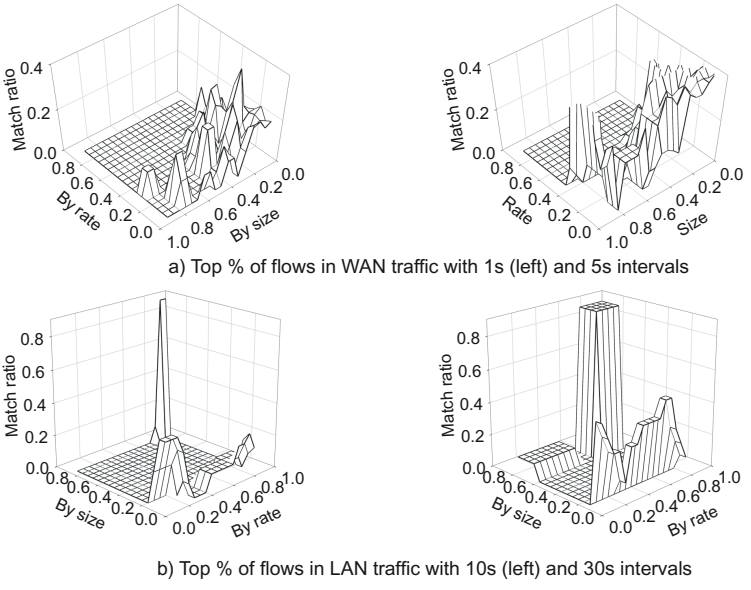


Fig. 8. Localized matches between threshold-selected flows ordered by size and rate

with the increase of the interval another feature gains strength and is positioned at the top of size list, but in the middle of rate list. Those are elephant flows, and the fact that they are found in the mid-area of rate axis supports the argument that large flows are not always fast. There are no matches when both size and rate lists are very short.

The LAN part of Fig.8 has the same features and additional peak at the area of short size and rate lists. In this case top flows by size really do match those by rate and the area of such matches grows with longer intervals. However, LAN plots also have features found in WAN, such as the fact that short size lists find good matches in lower parts of the rate list. The lack of feature when both lists are short in WAN can be potentially explained by high variability of size, which resulted in samples too scattered to fit in the localized matching window. However, larger windows will result in smoother/lower matching rates, which makes it almost impossible to verify whether there are even partial matches at the top of both lists. In any case, we can state that high levels of traffic aggregation result in higher variance of size, thus making the task of finding matches more difficult.

4 Conclusion

This paper proposed a model based on source interactivity that can explain high temporal volatility in identification of elephant flows. Based on practical results it is displayed that elephants are not always found at the top of the list ordered by size. The proposed source interactivity model explains this with short

traffic spikes coming from highly interactive (no feedback required) applications that can temporarily overwhelm the list of elephants flows. In this case, the real elephants transmit less traffic, but preserve transmission rate, which in our model is dependent on interactivity parameter. This means that identification of elephants by rate is more stable that that by size. This, however, is left for future research, as well as a more detailed parameterized interactivity model that would allow more rigid classification of traffic groups based on source interactivity, and, subsequently, on source behaviour.

Acknowledgments

This research is sponsored by KDDI R&D Laboratories Inc.

References

1. Claffy K., Braun H. and Polyzos G.C.: A parameterizable methodology for internet traffic flow profiling. IEEE Journal on Selected Areas in Communications 13(8) (1995) 1481–1494
2. Abrahamsson H. and Ahlgren B.: Temporal characteristics of large IP traffic flows. Technical Report T2003.27, Swedish Institute of Computer Science (2004)
3. Papagiannaki K., Taft N., Bhattacharyya S., Thiran P., Salamatian K. and Diot C.: A pragmatic definition of elephants in internet backbone traffic. 2nd ACM SIGCOMM Workshop on Internet Measurement (2002) 175–176
4. Papagiannaki K., Taft N., Bhattacharyya S., Thiran P., Salamatian K. and Diot C.: On the Feasibility of Identifying Elephants in Internet Backbone Traffic. Tech.Report no. RR01-ATL-110918, Sprint ATL (2001)
5. Papagiannaki K., Taft N. and Diot C.: Impact of Flow Dynamics on Traffic Engineering Design Principles. IEEE INFOCOM 4 (2004) 2295–2306
6. Crovella M.: Performance Evaluation with Heavy Tailed Distributions. Lecture Notes in Computer Science 2221 (2001) 1–11
7. Willinger W., Taqqu M., Sherman R. and Wilson D.: Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN traffic at the Source Level. IEEE/ACM Transactions on Networking 5(1) (1997) 71–86
8. Crovella M. and Taqqu M.: Estimating the Heavy Tail Index From Scaling Properties. Methodology and Computing in Applied Probability 1 (1999) 55–79
9. Erramilli A., Pruthi P. and Willinger W.: Self-Similarity in High-Speed Network Traffic Measurements : Fact or Artifact? VTT Symposium 154 (1995) 299–310
10. Zhang Y., Breslau L., Paxson V. and Shenker S.: On the characteristics and origins of internet flow rates. ACM SIGCOMM (2002) 309–322

A Profile Based Vertical Handoff Scheme for Ubiquitous Computing Environment

Chung-Pyo Hong, Tae-Hoon Kang, and Shin-Dug Kim

Dept. of Computer Science, Yonsei University, Seoul 120-749, Rep. of Korea
{hulkboy, thkang, sdkim}@yonsei.ac.kr

Abstract. Specifically ubiquitous computing paradigm is based on seamless connectivity. And also to guarantee the seamless connectivity, an intelligent network management between heterogeneous network interfaces, called vertical handoff (VHO), should be designed. Besides, conventional VHO schemes are based on network information only, and do not provide any service management scheme for seamless service. For this reason, this paper proposes a middleware architecture supporting efficient seamless connectivity with service management. In this architecture, we define modules to perform VHOs in accordance with contexts of services, user intention as well as network information. And also we define profiles used to express the contexts related to the process of VHOs and service management. With the proposed scheme, we can reduce the number of unnecessary VHOs and application failures significantly. This leads to around 130% of enhancement in data throughput, and also 85% of application failures can be reduced.

Keywords: Vertical handoff, seamless connectivity, application profile, context-aware processing, and ubiquitous computing.

1 Introduction

With the enormous practical potential of application management and the network manipulation method, ubiquitous computing has become a hot research area in computer science. In the ubiquitous computing, the most important assumption is that every component in a certain environment is connected to each other. So, seamless connectivity must be regarded as a major challenge for the ubiquitous computing. Specifically, in the area where heterogeneous network interface is available, it is recommended to utilize every network interface appropriately. The management scheme to move from one network interface to another is called vertical handoff (VHO).

Nevertheless, almost every research on VHO introduced recently is based on only network information, such as bandwidth, signal strength, and packet error rate. And they try to provide transparent handoff to service layer without considering other contexts like service quality and user intention. In the aspect of application services, any ubiquitous computing scheme not considering service quality management based on network status or heterogeneous network environment leads to the failure of application services. This may corrupt the continuity or stability of user intended work.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 102 – 111, 2006.
© Springer-Verlag Berlin Heidelberg 2006

To overcome this problem, we propose a middleware approach. The proposed middleware supports a dynamic service quality management based on the network information in the upper layer and efficient VHO with the contexts representing the service requirement or user intention as well as network information in the lower layer. In this architecture, an application agent supporting the management of service quality as network status varies, specially designed VHO decision module, and other necessary modules are introduced. And also, we define some profiles related to the context of applications services, user intention, and network status. Through the middleware architecture and profiles we defined, this paper provides an efficient VHO scheme with dynamic service management, needed for seamless service, based on not only network information but also profiles reflecting rich contexts of user intention and environment. In the simulation results, the throughput improvement obtained by the reduced number of unnecessary VHOs shows about 130% enhancement, and the number of application failures related to the continuity of services shows about 85% decrease.

The remaining part of this paper consists of four parts. In Section 2, related work is introduced. In Section 3, proposed schemes, middleware architecture, and algorithms, are presented. Section 4 shows evaluation result and, lastly, conclusions are presented in Section 5.

2 Related Work

In this section, some related research is described. GAIA is one of the most famous ubiquitous frameworks. It provides dynamic services based on profiles generated by environmental contexts, e.g., network information, user moving information, and environmental brightness. In GAIA, they suggest a scheme handling QoS when network connection status is changed, but they did not mention how to control network status or service quality. They just assume that every device in heterogeneous network environments can be connected each other [1]. Aura, another famous framework, is a good example, too. The aspect they are interested in is not to utilize network interfaces but to manage services dynamically in a single network environment [2].

In VHO scheme, a location service server (LSS) to provide mobile terminals with the information about nearby networks is introduced in [3], where the VHO decision scheme is based on the network oriented information. Thus, applications passively receive the consequence of vertical handoff process. This mechanism causes some unnecessary vertical handoff processes. The integration of WLAN and GSM/GPRS with multi-tunnel scheme is proposed in [4]. This scheme uses dwell timers and thresholds to perform the VHO properly. Although it represents QoS oriented scheme, it does not consider any application requirement. In [6] a system for integration of 802.11 and 3G networks is proposed. The system is constructed in loosely coupled IOTA (Integration Of Two Access technologies) gateway and IOTA client software. IOTA gateway manages the network interfaces and prevents network oscillation, but it only concerns network factors, such as signal strength, priority, threshold, and so on. In [7], a new handoff protocol for overlay networks, HOPOVER (Handoff Protocol for Overlay Networks), is proposed. HOPOVER enables smooth handoffs

intra- and inter-network through three procedures, pre-resource reservation, buffering and forwarding, but it only focuses on BSs (base stations). In [8], a vertical handoff system based on Wireless Overlay Networks - a hierarchical structure of room-size, building-size, and wide area data networks - is proposed. This vertical handoff system gives mobile devices the ability to roam freely in wireless overlay networks. But, this system only concerns about tradeoff between latency and cost. In [4][6][7][8], all proposed schemes consider only network factors under the condition of handoff among heterogeneous networks, and do not concern application requirements.

A good example is proposed in [5]. It describes active application oriented (AAO) scheme which performs VHO with application requirements. It performs VHO with application requirements rather than the issues of network layer, but AAO scheme assumes that only one application runs on mobile terminal. The QoS factor of AAO scheme is represented by quantized value of overall network condition. Thus, QoS problem on each factor required by any specific application is not considered. And also in [5], they do not consider other methods for service continuity when there is not any adequate network interface.

3 Proposed Architecture

We design a profile-based middleware architecture supporting seamless connectivity for ubiquitous computing. The proposed architecture is to use a specially designed application profile and a profile named as working profile which represents user intention. With the working profile and the application profile, the proposed architecture can perform the VHO and the application service management at the right time. In this way, we can improve the accuracy to initiate VHO and get the best performance in network utilization. In this section, the newly designed modules and the profiles are described in detail.

3.1 Proposed Profiles

The principal architectural goal in designing proposed architecture is to achieve the following two issues. The first is to provide users with the environment for efficient work progress. The second is to obtain the best performance gain even though user moves around the region where the network status varies. To achieve these two goals, we define a working profile and an application profile. As shown in Table 1, the working profile is a set of names and policies of applications collected for any particular intended work. The working profile is to be generated by the work conductor, which is a kind of rule-based context engine inferring user intention based on information from various contexts [12]. Policy is decided by a work conductor following the user intention. The policy M means that the application is a mandatory application to execute the work and R is a recommended application.

Table 1. An example of an working profile : a business trip

Application Name	Work Policy
Navigation	M
Stock information ticker	R

The application profile is also defined, where service mode description is a key feature. Each service mode has different network factors. It represents the multi modal service of an application. As network environment is changed, the application can change its service mode based on the application profile. An example of the application profile is presented in Table 2.

3.2 Application Management Middleware Architecture

The proposed application management architecture consists of an application manager, a VHO decision manager, a working profile manager, an application agent, and a work conductor. Basically the application manager aggregates all the requirements of services performed by applications. It also makes the abstracted profile which is generated from all the requirements of multiple applications. The abstracted profile is sent to the VHO decision manager after it is generated. The key role of the VHO decision manager is to compare the abstracted profile with current network information and to decide whether it is necessary to perform handoff or not. When there is not a proper network interface to switch, the VHO decision manager tries to find an adequate network interface and performs the handoff operation. A detailed description of the VHO decision manager is presented in Section 3.4. The working profile manager manages the working profile. It supplies the working profile to the application agent. And the application agent manages application service modes and requests the work conductor to reconfigure the intended work. The work conductor manages how to invoke, suspend, and change service mode for any given application, and also generates a working profile for user intended work. Fig 1. is the overall architecture of the proposed architecture.

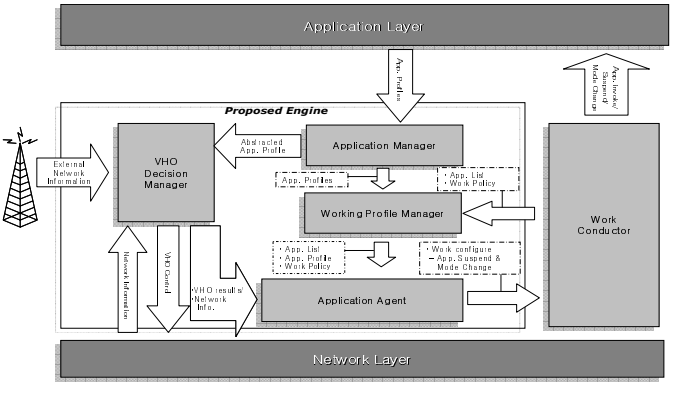


Fig. 1. Overall architecture of the proposed middleware

3.3 Overall System Operational Flow

A detailed explanation is followed as shown in Fig 2. In step 1, a work is invoked. The work conductor invokes all the applications for a certain work and the applications are running through step 2. The application profiles are passed to the step 3. In step 3, an abstracted profile is generated from the application profiles. In step 4,

working profile is generated and passed to step 7, which facilitates the applications service mode change. The 5th step performs the VHO decision and if necessary, finds another proper network. Step 6 is the actual step to perform the VHO and in a certain case it calls the application agent. The application agent manages the service mode in the step 7 in a certain situation.

Table 2. An example of application profile

Application Name	Policy	Service Mode			
		Mode		Min Value	Max Value
Navigation	M	Mode1 : Video based	Bandwidth (Kbps)	30	128
			Packet Error Rate (%)	3	1
			Latency (ms)	120	5
		Mode2 : Audio based	Bandwidth	9.6	64
			Packet Error Rate	20	1
			Latency	8	2

3.4 Vertical Handoff Decision Manager

VHO decision manager decides whether it should perform any handoff and requests the network layer to change any chosen network interface if necessary, based on the abstracted profile and network information. The abstracted profile is generated by the application manager and needed to specify any given network environment supporting all the applications running on any mobile terminal. In this paper, we assume that network layer provides the information about other available network interfaces as well as current network interface through control channel.

3.4.1 Application Profiles Abstraction

The application manager gathers profiles of the applications running on any mobile terminal and keeps those in its repository. An example of running application profile is shown in Table 3 and its associated abstracted profile obtained from Table 3 is shown in Table 4. The application profile consists of some network factors, such as bandwidth and network latency based on current service mode. For the n-th network factor, denoted by *fn*, an application profile defines boundaries for that factor and each factor is described as upper bound, *Ufn* and lower bound, *Lfn*. These expressions are mentioned in [5], and we borrow it in our proposed scheme.

Based on profiles stored in the repository, an abstracted profile is generated. The process of making abstracted profile is quite simple. In the case of *high factor*, the biggest values of *Ufn*s and *Lfn*s are chosen as the abstracted factor of profiles denoted as *AUfn* and *ALfn*. On the contrary, in the case of *low factor*, the smallest values of *Ufn*s and *Lfn*s are chosen as the abstracted factor of profiles. High factors are factors with value that should be as high as possible and low factors are factors with value that should be as low as possible [5]. The newly generated factors like *AUfn* and *ALfn* constitute abstracted profile and it is sent to VHO decision manager. In this paper, we do not consider any extra bandwidth required to process applications simultaneously.

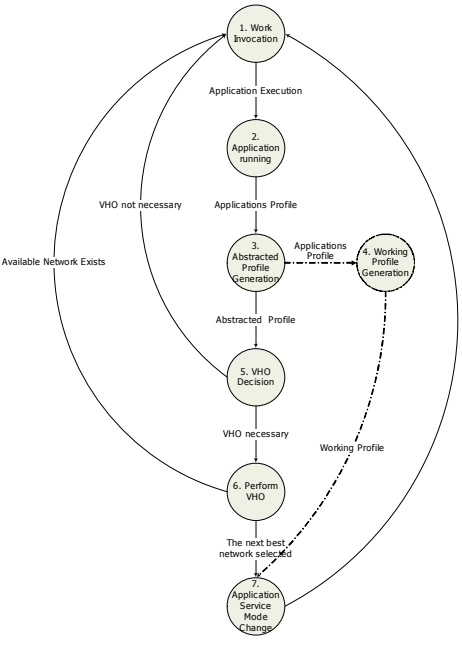


Fig. 2. Overall flow state diagram

Table 3. An example of running application profile

Application Name	Policy	Service Mode			
		Mode		Min Value(*Lfn*)	Max Value(*Ufn*)
Navigation	M	Mode1 : Video based	Bandwidth (Kbps)	128	500
			Packet Error Rate (%)	8	2
			Latency (ms)	20	5
Stock Info. Ticker	R	Mode4 : Text based	Bandwidth	9.6	64
			Packet Error Rate	8	2
			Latency	20	1

Table 4. An example of abstracted profile from Table 3

	Abstracted Profile	
	ALfn	*AUfn*
Bandwidth (Kbps)	128	500
Packet error rate (%)	3	1
Latency (ms)	20	1

3.4.2 Network QoS Measurement

The most important thing before performing VHO is to verify the QoS of current network environment. In this paper, we will use normalized factors based on the abstracted profile of factors in network interface *m*. We can get high and low

normalized factors from equation (1) and (2). These equations are modified from the expression in [5].

High factor normalization :

$$f_{m,n} = \begin{cases} -1, \frac{UR_{m.n} + LR_{m.n}}{2} < ALf_n \\ 0, ALf_n \le \frac{UR_{m.n} + LR_{m.n}}{2} \le AUf_n \\ 1, \frac{UR_{m.n} + LR_{m.n}}{2} > AUf_n \end{cases} . \quad (1)$$

Low factor normalization :

$$f_{m,n} = \begin{cases} -1, \frac{UR_{m.n} + LR_{m.n}}{2} > ALf_n \\ 0, ALf_n \ge \frac{UR_{m.n} + LR_{m.n}}{2} \ge AUf_n \\ 1, \frac{UR_{m.n} + LR_{m.n}}{2} < AUf_n \end{cases} . \quad (2)$$

URm,n and *LRm,n* represent the range of real values of factor *fn* in the network *m* and *fm,n* represents normalized factor in network *m*.

Normalized factors are updated periodically. VHO decision is activated when more than one of average normalized factor's value per system defined time is lower than zero. In this manner, each factor can be considered more carefully. Thus, more precise VHO time and QoS oriented verification can be acquired [11][13].

3.4.3 Handoff Decision Algorithm

In most of VHO schemes, VHO algorithm is managed by network layer. Applications do not consider any VHO process. This paper proposes a scheme that considers application profile in the first step. Application manager generates the abstracted profile with abstracted factors. VHO decision manager generates normalized factors and updates those periodically. When more than one of average factor's value per system defined time is lower than zero, VHO decision manager attempts to find nearby air interfaces available. As maintaining current network, VHO decision manager verifies new network interfaces in dwelling time if there exist nearby networks. The dwelling time in the proposed scheme is defined as 60 seconds. It is defined as a time to move one kilometer if mobile terminal moves at 60km/h. If not, mobile terminal stays at current network. When one of new interfaces is confirmed to be steady, VHO decision manager decides to switch to a new network and request the application agent for the service mode change. We decided to control the service mode whenever VHO occurs because the new network interface may guarantee the better QoS level for the higher service modes.

4 Evaluation

In this section, we present simulation results to compare our middleware architecture with others. Single application based model without service mode management such as AAO (active application oriented) scheme, single application based model which

supports service mode management, and multi application supporting scheme which do not support the service mode management will be employed to be compared.

4.1 Simulation Model

In this section, a scenario based simulation methodology is presented. In this scenario, user is defined as a stock broker. And the mobile terminal assumed is a dual band device. GSM/GPRS and WLAN are deployed on the mobile terminal. When the terminal wakes up, it starts with GSM.

The scenario consists of three parts. Each part corresponds to the work we defined. The first work is about a way to office or home. The way to office starts at 7:00 and finishes at 8:00 and the way to home starts at 18:00 and 20:00. In this case, working profile consists of navigation (M) and broadcasting application(R). (M) and (R) indicate the mandatory and recommended applications. The second work is to move to the restaurant for lunch. The user wants the way to the restaurant to be informed and wants to work on his stock trading as moving to the restaurant. So, in this case, the user uses applications of navigation, restaurant information, stock trade and stock information ticker. All the applications used in this case are recommended. And the final work is an accidental work occurrence. In this paper, we assume that the accidental work is consists of two applications. One is data transmission (R) and another is communication (M). This accidental work arises at any time between 7:00 and 20:00 and it carries on from one hour to three hours. The third work can occur even when other work is processed and it is the dominant work to others. An example of the working profile is mentioned in Table 1.

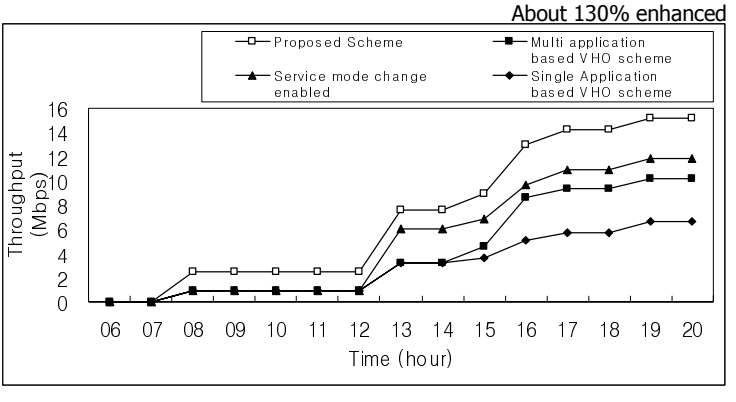


Fig. 3. Accumulated throughput comparison

4.2 Simulation Result

When discussing the result, we use the total throughput generated by the simulation to compare several methods. Because it denotes the total amount of information needed to perform a certain user intended work. And also, we use the measure of the application failures number as the reason that may cause any problem on the continuity of the user intended work.

As shown in Fig. 3, the throughput of the multi application based model is higher than single application based model by 55%. And the service mode enabled model offers the 79% higher throughput than single application based model. Finally, the throughput tends to increase by around 130% with the proposed scheme.

In Fig. 4, the number of application failures is presented. As shown in Fig. 4, the number of application failures is the smallest when we apply the proposed scheme. And the conventional scheme which is based on the single application oriented method has the highest number of application failures. The number of application failures for the conventional method is decreased by 85%, comparing with that of the proposed method. It represents that the proposed scheme is an efficient method to utilize network environment.

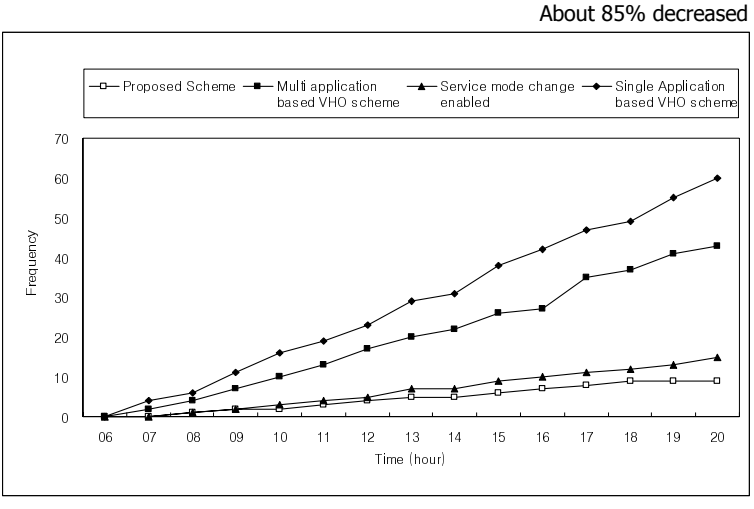


Fig. 4. Application failures number comparison

5 Conclusion

An efficient network management in ubiquitous computing is very important to provide seamless service. And also, in heterogeneous network environment, a management scheme should be designed to utilize every network interface properly. This kind of network management scheme is called as VHO. Conventional VHO schemes are built based only on an idea representing the transparent handoff between heterogeneous network interfaces. And it is considered to be a different research area to the seamless service management, so any way to provide seamless services to users is not considered. For this reason, we design a middleware architecture with VHO decision module, application agent, and any necessary modules. In this architecture, we define several profiles reflecting various contexts required for seamless service. Whit this architecture we can achieve around 130% of performance enhancement in data throughput and around 85% of application failures can be reduced compared with the conventional schemes.

Reference

1. Román, M., Hess, C. K., Cerqueira, R., Ranganathan, A., Campbell, R. H., Nahrstedt, K.: Gaia: A Middleware Infrastructure to Enable Active Spaces. IEEE Pervasive Computing, Vol. 1. (2002) Page(s):74-82.
2. Sousa, J. P., Garlan, D.: Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments. Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture (2002) Page(s):29-43.
3. Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M.: MIRAI: A Solution to Seamless Access in Heterogeneous Wireless Networks. ICC2003, Vol. 2. (2003) Page(s):1033-1037.
4. Ye Min-hua, Liu Yu and Zhang Hui-min: The mobile IP Handoff Between Hybrid Networks. PIMRC2002, Vol. 1. (2002) Page(s):265-269.
5. Chen, W. T., Shu, Y. Y.: Active Application Oriented Vertical Handoff in next-Generation Wireless Networks. WCNC2005, Vol. 3. (2005) Page(s):1383-1388.
6. Buddhikot, M., Chandranmenon, G., Han, S., Lee, Y. W., Miller, S., Salgarelli, L.: Integration of 802.11 and Third-Generation Wireless Data Networks. INFOCOM 2003, vol. 1. (2003) Page(s):503-512.
7. Du, F., Ni, L. M., Esfahanian, A. H.: HOPOVER: A New Handoff Protocol for Overlay Networks. ICC 2002, vol. 5. (2002) Page(s):3234-3239.
8. Stemm, M., Katz, R. H.: Vertical Handoffs in Wireless Overlay Networks: ACM Mobile Networking (MONET). Special Issue on Mobile Networking in the Internet 1998. Vol. 3. (1998) Page(s):335-350.
9. IEEE Standards Association, 802.11b Standard, http://standards.ieee.org/
10. GSM World, GPRS-Standard class 10, http://www.gsmworld.com/
11. C. P. Hong., T. H. Kang., and S. D. Kim.: An Effective Vertical Handoff Scheme Supporting Multiple Applications in Ubiquitous Computing Environment. The 2nd International Conference on Embedded Software and Systems (2005) Page(s):407-412
12. Kwang-Won Koh., Chang-Ik Choi., Kyung-Lang Park., Shin-Young Lim., and Shin-Dug Kim.: A Multilayered Context Engine for the smartHome. International Conference on Computer science, Software engineering, Information Technology, E-business, and Applications (2004)

The Soft Bound Admission Control Algorithm for Vertical Handover in Ubiquitous Environment

Ok Sik Yang[1], Jong Min Lee[1], Jun Kyun Choi[1], Seong Gon Choi[2], and Byung Chun Jeon[3]

[1] Information and Communications University (ICU), 119 Munji-Dong, Yuseong-Gu, Daejeon 305-732, Republic of Korea
[2] Chungbuk National University, 12 Gaeshin-Dong, Heungduk-Gu, Korea
[3] Netvision Telecom, Tamnip-dong, Yuseong-gu, Daejeon Korea
yos@icu.ac.kr

Abstract. In this paper, we present SBAC (Soft Bound Admission Control) algorithm considering critical bandwidth ratio to reduce handover blocking probability over WLAN and WAAN (Wide Area Access Network). SBAC algorithm utilizes dynamically optimized resource allocation scheme to decrease the blocking probability of vertical handover connections within the limited capacity of system. Based on SBAC algorithm, we derive the handover blocking probability as new traffic load and handover traffic load increase. In order to evaluate the performance, we compare SBAC algorithm against traditional non-bounded and fixed bound schemes. Numerical results show that the SBAC scheme improves handover blocking probability in ubiquitous environment.

1 Introduction

In recent, the internet is expected to support bandwidth intensive services along with traditional modes of data traffic with the increase of wireless devices (e.g. 3G cellular, WLAN, Bluetooth). Furthermore, due to the increase of mobile users and environmental limitation, current mobile networks need mechanisms to efficiently handle the resource management for seamless handover in ubiquitous environment. In such environment, a users or network will be able to decide where to handover among the different access technologies based on the bandwidth, cost, and user preferences, application requirements and so on. Therefore, efficient radio resource management and connection admission control (CAC) strategies will be key components in such a heterogeneous wireless system supporting multiple types of applications with different QoS requirements [1].

Many admission control schemes have been proposed to enable the network to provide the desired QoS requirements by limiting the number of admitted connections to that network to reduce or avoid connection dropping and blocking [2], [3]. In ubiquitous environment, other aspects of admission control need to be considered due to handover. If the wireless network is unable to assign a new channel due to the lack of resources, an accepted connection may be dropped before it is terminated as a result of the mobile user moving from its current place to another during handover. Since dropping an ongoing connection is generally more sensitive to a mobile user than blocking a new connection request, handover connections should have a higher priority over the new

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 112 – 121, 2006.
© Springer-Verlag Berlin Heidelberg 2006

connections in order to minimize the handover blocking probability. On the other hand, reducing the blocking of handover connection by channel reservation or other means could increase blocking for new connections. There is therefore a trade off between these two QoS measures [4]. The problem of maintaining the service continuity and QoS guarantees to the multimedia applications during handover is deteriorated by the increase of vertical handover in heterogeneous wireless networks.

In the ubiquitous environment, vertical handover considering user preferences, traffic characteristic, user mobility range, and so on could occur more frequently than horizontal handover. Therefore, vertical handover should have higher priority to support QoS requirement because it considers more various factors (e.g. cost, bandwidth, velocity, etc.) than horizontal handover. So we proposed a dynamic admission control for vertical handover connections in ubiquitous environment.

This paper is organized as follows. In the next section, we describe the architecture of proposed algorithm. In section 3, we propose a soft admission control algorithm using softness profile. Numerical results obtained using the traditional methods are presented and compared in section 4. Finally, we conclude the paper in section 5.

2 The Architecture of Proposed Algorithm

2.1 Network Architecture

Fig. 1 shows the network architecture for mobility service in ubiquitous environment. This architecture is based on IPv6 networks to support the movement of every user.

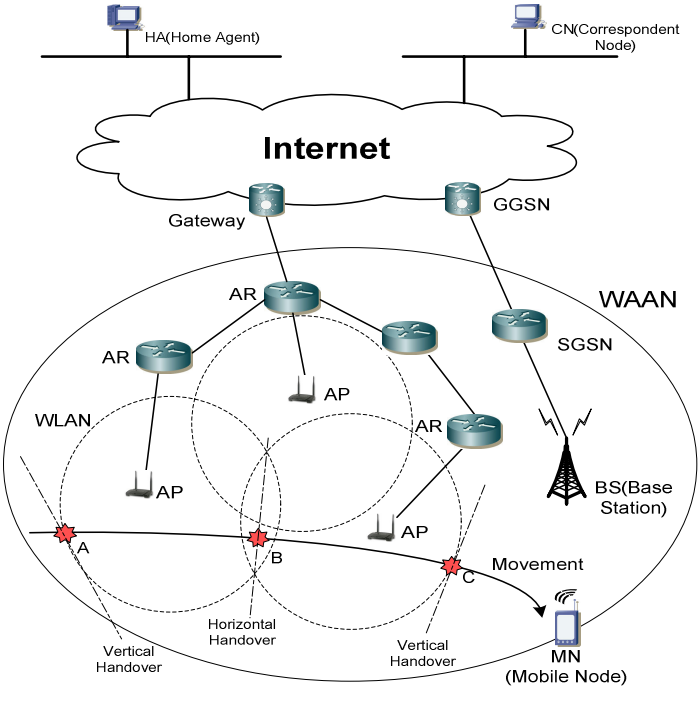


Fig. 1. Network architecture for mobility service in ubiquitous environment

There are two kinds of users that can handover under this architecture. One is WLAN to WAAN users and vice versa. Therefore, there will be two handovers between WLAN and WAAN: vertical handover and horizontal handover.

We assume that a user has multi-interface terminal [5]. As shown in Fig. 1, the connection initiated in WAAN and mobile node is moving to right side. When the mobile node jumps into another access area, it requires vertical or horizontal handover connection to continue their movement or connection, where A, B and C are handover points respectively. A loosely coupled inter-working approach can be considered for implementation. In this case, Mobile IP [6] mechanism must be implemented in all equipment including MNs (Mobile Nodes). This approach also should support more than one IP address for one mobile user so that one user can access more than one wireless system simultaneously. Finally, on the top of a network, a suitable resource allocation mechanism is required to control the traffic and system load. SBAC algorithm is exploited here in this architecture [7].

3 Proposed SBAC Algorithm

3.1 Softness Profile

The softness profile is defined on the scales of two parameters: satisfaction index and bandwidth ratio [8]. The satisfaction index is a mean-opinion-based (MOS) value graded from 1 to 5, which is divided by two regions: the acceptable satisfaction region and low satisfaction region.

Bandwidth ratio graded from 0 to 1 can be separated by 3 regions. In the region from 1 to A, it has low degradation of satisfaction index. It means users are not sensitive in this region. However, it has large degradation of satisfaction index in the region from A to B.

The point indicated as B is called the critical bandwidth ratio (ξ) used in proposed algorithm. Since this value is the minimum acceptable satisfaction index, it can be threshold of bandwidth ratio. In the region from B to 0, users do not satisfy their services. Therefore, this region should not be assigned to any users. Generally, the critical bandwidth ratio (ξ) of Video On Demand (VOD) is 0.6 ~ 0.8 and back ground traffic is 0.2~0.6 [9].

3.2 Proposed SBAC Algorithm

Proposed SBAC algorithm is illustrated in Fig. 2. This algorithm shows dynamically optimized bound handover procedure within given total bandwidth B_{total}.

When a mobile node requires bandwidth for new or handover connections, mobile agent checks the available bandwidth within some threshold to decide connection admission or rejection. In this point, handover connections should be treated differently in terms of resource allocation. Since users tend to be much more sensitive to connection dropping (e.g. disconnection during VoD service) than to connection blocking (e.g. fail to initiate connection), handover connections should assign higher priority than the new connection. Especially, the vertical handover connection needs to have higher priority than horizontal handover connection because it considers more

various factors (e.g. cost, bandwidth, velocity, etc.) than horizontal handover connection. In this time, if there is no available bandwidth to accept vertical handover connection, mobile agent calculates the optimized critical bandwidth ratio. If it is bigger than threshold based on softness profile, mobile agent reassigns bandwidth based on decided critical bandwidth ratio (ξ). As a result, the vertical handover connections can be accepted more than horizontal handover connections.

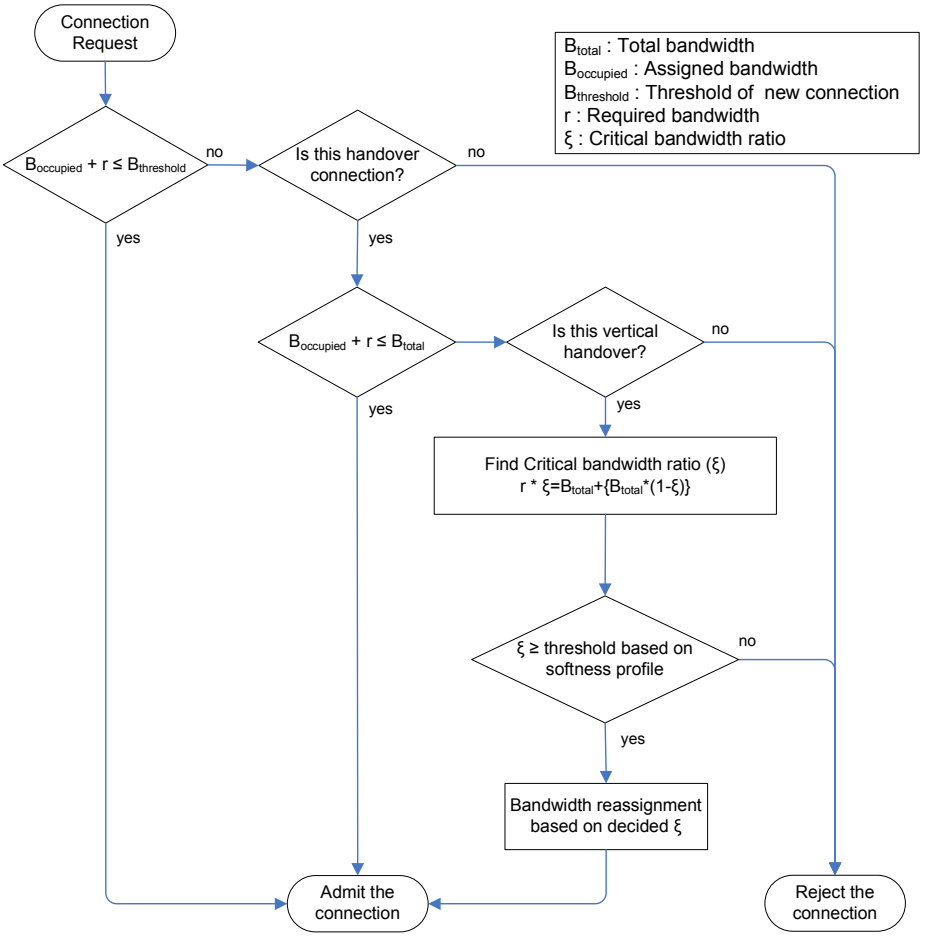


Fig. 2. Proposed SBAC algorithm

Fig. 3 shows the system model of the proposed approach. Let C denote the total capacity and P_{nb}, P_{hb}, P_{vb} denote the blocking probabilities of new connection, horizontal handover and vertical handover connection respectively. The arrival process of new and handover connections is assumed to be Poisson and denoted by λ_n and $\lambda_h + \lambda_v$. The factor $\alpha = \lfloor (1-\xi) * C \rfloor$ is bandwidth donated from each user

within the same coverage without critical degradation of service. From the assumption, we obtain the blocking probability of horizontal handover connection, vertical handover connection, and new connection.

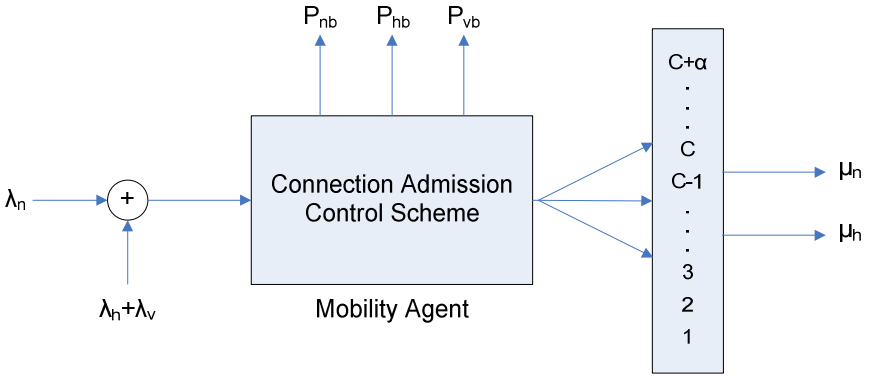


Fig. 3. System model

Fig. 4 indicates the transition diagram for the proposed scheme. The detailed notations used in this diagram are shown in Table. 1.

Table 1. Notations

Notation	explanation
λ_n	Arrival rate of new connection
λ_h	Arrival rate of horizontal handover connection
λ_v	Arrival rate of vertical handover connection
$1/\mu_h$	Average channel holding time for handover connections
$1/\mu_n$	Average channel holding time for new connections
C	Maximum number of server capacity
T	Threshold (bound of the total bandwidth of all accepted new connections)
ξ	Critical bandwidth ratio
α	$\lfloor (1-\xi)*C \rfloor$
n_n	Number of new connections initiated in the coverage
n_{hv}	Number of handover connections in the coverage

In order to analyze the blocking probability of each connection, we use the two-dimensional Markov chain model with the state space S and M/M/C+α/C+α [10] model is utilized.

$$S = \{(n_n, n_{hv}) | 0 \le n_n \le T, (n_n + n_{hv}) \le C + \alpha\} \tag{1}$$

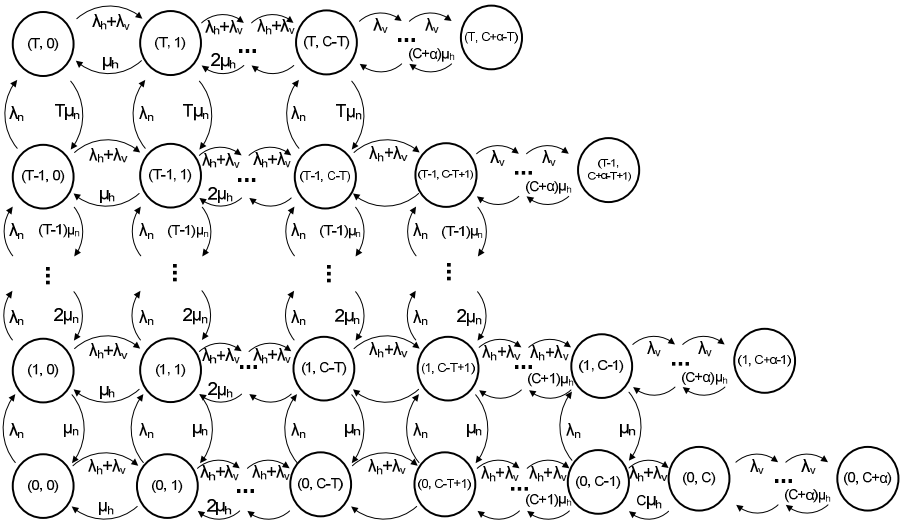


Fig. 4. Transition diagram for the proposed SBAC algorithm

Let $q(n_n, n_{hv}; \overline{n_n}, \overline{n_{hv}})$ denote the probability transition rate from state (n_n, n_{hv}) to $(\overline{n_n}, \overline{n_{hv}})$. Then, we obtain the below.

$$\begin{aligned}
&q(n_n, n_{hv}; n_n + 1, n_{hv}) = \lambda_n && (0 \le n_n < T, 0 \le n_{hv} \le C)\\
&q(n_n, n_{hv}; n_n - 1, n_{hv}) = n_n \mu_n && (0 < n_n \le T, 0 \le n_{hv} \le C)\\
&q(n_n, n_{hv}; n_n, n_{hv} + 1) = \lambda_h + \lambda_v && (0 \le n_n \le T, 0 \le n_{hv} < C)\\
&q(n_n, n_{hv}; n_n, n_{hv} - 1) = n_{hv} \mu_h && (0 \le n_n \le T, 0 < n_{hv} \le C)\\
&q(n_n, n_{hv}; n_n, n_{hv} + 1) = \lambda_v && (0 \le n_n \le T, C \le n_{hv} < C + \alpha)\\
&q(n_n, n_{hv}; n_n, n_{hv} - 1) = n_{hv} \mu_h && (0 \le n_n \le T, C < n_{hv} \le C + \alpha)
\end{aligned} \tag{2}$$

Let $p(n_n, n_{hv})$ denote the steady-state probability that there are new connections (n_n) and vertical and horizontal handover connections (n_{hv}). By using the local balance equation [10], we can obtain

$$\begin{aligned}
&p(n_n, n_{hv}) = p(0,0) \cdot \frac{{\rho_n}^n \cdot {\rho_h}^{hv}}{n_n! \, n_{hv}!}\\
&\qquad where \ \rho_n = \frac{\lambda_n}{\mu_n}, \rho_h = \frac{\lambda_h + \lambda_v}{\mu_h}, (0 \le n_n \le T, 0 \le n_{hv} < C)\\
&p(n_n, n_{hv}) = p(0,0) \cdot \frac{{\rho_n}^n \cdot {\rho_h}^C \cdot \rho^{hv-C}}{n_n! \, n_{hv}!}\\
&\qquad where \ \rho = \frac{\lambda_n}{\mu_n}, \rho = \frac{\lambda_v}{\mu_h}, (0 \le n_n \le T, C \le n_{hv} \le C + \alpha)
\end{aligned} \tag{3}$$

From the normalization equation, we also obtain

$$p(0,0) = \left[\sum_{0 \le n_n \le T, n_n + n_{hv} \le C+\alpha} \frac{\rho_n^{n_n}}{n_n!} \cdot \frac{\rho_h^{n_{hv}}}{n_{hv}!} \right]^{-1}$$
$$= \left[\sum_{n_n=0}^{T} \frac{\rho_n^{n_n}}{n_n!} h \cdot \sum_{n_{hv}=0}^{C-n_n} \frac{\rho_h^{n_{hv}}}{n_{hv}!} + \sum_{n_n=0}^{T} \frac{\rho_n^{n_n}}{n_n!} \cdot \sum_{n_{hv}=C}^{(C+\alpha)-n_n} \frac{\rho_h^{n_{hv}} \cdot \rho_h^{(n_{hv}-C)}}{n_{hv}!} \right]^{-1} \quad (4)$$

From this, we obtain the formulas for new connection blocking probability and handover connection blocking probability as follows:

$$P_{nb} = \frac{1}{P(0,0)} \cdot \sum_{n_{hv}=0}^{C-T} \frac{\rho^T}{T!} \cdot \frac{\rho^{n_{hv}}}{n_{hv}!} + \frac{1}{P(0,0)} \cdot \sum_{n_n=0}^{T-1} \frac{\rho^{n_n}}{n_n!} \cdot \frac{\rho_h^{C-n_n}}{(C-n_n)!}$$
$$P_{hb} = \frac{1}{P(0,0)} \cdot \sum_{n_n=0}^{T} \frac{\rho^{n_n}}{n_n!} \cdot \frac{\rho_h^{C-n_n}}{(C-n_n)!} \quad (5)$$
$$P_{vb} = \frac{1}{P(0,0)} \cdot \sum_{n_n=0}^{T} \frac{\rho^{n_n}}{n_n!} \cdot \frac{\rho_h^{(C+\alpha)-n_n}}{((C+\alpha)-n_n)!}$$

Obviously, when T is equal to C, the new connection bounding scheme becomes the non-prioritized scheme. As we expect, we can obtain [3]

$$P_{nb} = P_{hb} = P_{vb} = \frac{\frac{(\rho_n \rho_h)^C}{C!}}{\sum_{n=0}^{C} \frac{(\rho_n \rho_h)^n}{n!}} \quad (6)$$

4 Numerical Results

In this section, we present the numerical results for the comparison of performance. We compared three bounding schemes: non-bound, fixed bound, and SBAC algorithms.

In Fig. 5, we increase the handover connection traffic load (ρ_h). This graph shows the blocking probability of handover connection under the following parameters: C=35, T=20, λ_n = 1/20, λ_h =1/60, λ_v =1/60, μ_n =1/300, μ_h is varying from 1/100 to 1/1000, and ξ =0.9.

In this case, traffic load of handover connections (ρ_h) are increasing from 0 to 40 and traffic load (ρ_n) of the new connection is 15. Since handover connections are not bounded, as increasing the handover traffic load, the differences among three schemes become similar.

Fig. 6 shows handover blocking probability under the following parameters: C=35, T=15, λ_n = 1/30, λ_h =1/60, λ_v =1/60, μ_h =1/450, μ_n is varying from 1/100 to 1/800, and ξ =0.9.

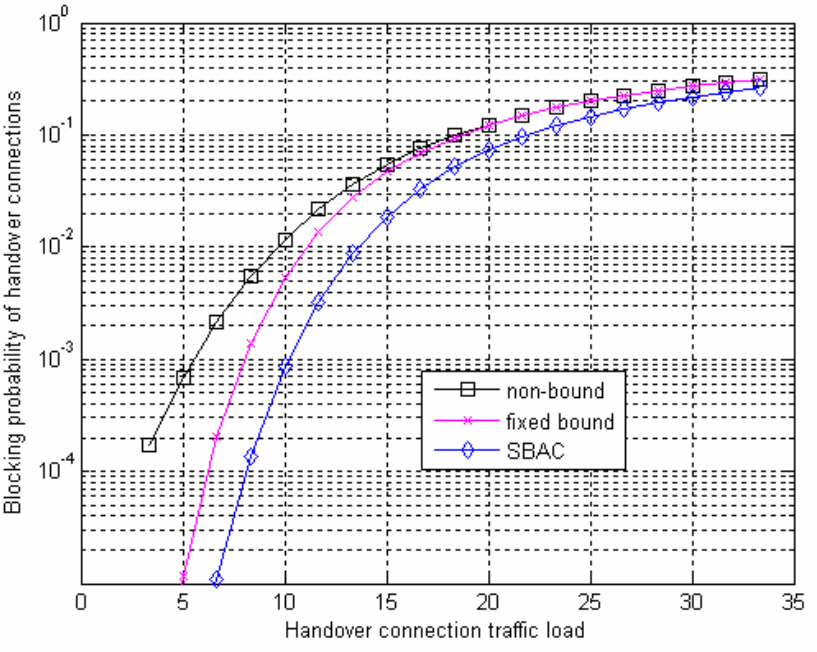


Fig. 5. Blocking probability of handover connections vs. handover connection load

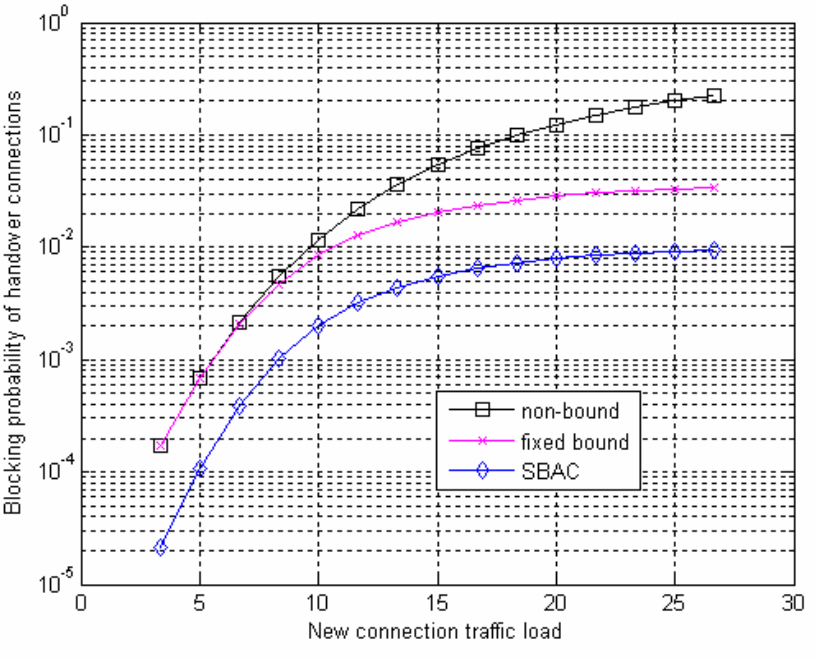


Fig. 6. Blocking probability of handover connections vs. new connection load

In Fig. 5 handover connection traffic load is given as ρ_h=15. It is observed that when traffic load of the handover connection is higher than the new connection traffic load (e.g. $\rho_h > \rho$), non-bound scheme and fixed bound scheme are not much different. On the other hand, when traffic load of the handover connection is lower than the new connection traffic (e.g. $\rho_h < \rho$), their blocking probabilities become different. This is the case when the new connections arrive in bursts (say after finishing a class) [11]. Our proposed SBAC algorithm can offer the seamless handover for ongoing connections with lower blocking probability.

In summary, the SBAC algorithm could achieve better results than the traditional bounding schemes, especially for the handover connection. These results can be used for analysis and design of ubiquitous environment.

5 Conclusion

In this paper, we proposed SBAC algorithm that reduces the blocking probability of vertical handover connections within the limited capacity of system over ubiquitous environment (e.g. WLAN and WAAN). Proposed SBAC algorithm considers vertical handover connections that have higher priority. In order to analyze the blocking probability of SBAC algorithm, we use the two-dimensional Markov chain model. From the numerical analysis, we compared SBAC algorithm against traditional non-bound and fixed bound scheme. As a result, proposed SBAC scheme is able to improve handover blocking probability in ubiquitous environment. Future work needs to analyze the utilization based on optimized critical bandwidth ratio.

Acknowledgement. This work was supported in part by the MIC, Korea under the ITRC program supervised by the IITA and the KOSEF under the ERC program.

References

1. Dusit Niyato, Ekram Hossain: Call Admission Control for QoS Provisioning in 4G Wireless Networks: Issues and Approaches, IEEE Network, Sept.-Oct. (2005) 5-11
2. I. Katzela, M. Naghshineh: Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey, IEEE Personal Commun., vol. 3, June (1996) 10–31
3. Yuguang Fang, Yi Zhang: Call Admission Control Schemes and Performance Analysis in Wireless Mobile Networks, IEEE Transactions on Vehicular Technology, Vol. 51, No. 2, March (2002) 371-382
4. Emre A. Yavuz, Victor C. M. Leung: A Practical Method for Estimating Performance Metrics of Call Admission Control Schemes in Wireless Mobile Networks, IEEE WCNC, March (2005) 1254-1259
5. M. Buddhikot et al.: Design and Implementation of a WLAN/CDMA2000 Interworking Architecture, IEEE Communications Magazine, Nov. (2003)
6. D. Johnson, C. Perkins and J. Arkko,: Mobility Support for IPv6, RFC 3775, June (2004)
7. Sheng-Tzong Cheng, Jian-Liang Lin: IPv6-Based Dynamic Coordinated Call Admission Control Mechanism Over Integrated Wireless Networks, IEEE Journal on Selected Areas in Communications, Vol. 23, No. 11, Nov. (2005) 2093-2103

8. Reininger D., Izmailov R.: Soft quality-of-service control for multimedia traffic on ATM networks, Proceedings of IEEE ATM Workshop, (1998) 234-241
9. Sung H. Kim, Yeong M. Jang: Soft QoS-Based Vertical Handover Scheme for WLAN and WCDMA Networks Using Dynamic Programming Approach, LNCS 2524, Nov. (2002) 707-716
10. Kleinrock. L.: Queueing System, Vol. 1 Theory, John Wiley and Sons, New York (1975)
11. Jiongkuan Hou, Yuguang Fang: Mobility-based call admission control schemes for wireless mobile networks, Wireless Communications and Mobile Computing. Wirel. Commun. Mob. Comput. (2001) 1:269–282

Improving Handoff Performance by Using Distance-Based Dynamic Hysteresis Value

Huamin Zhu and Kyungsup Kwak

Graduate School of Information Technology and Telecommunications
Room 428, High-Tech Center, Inha University
#253 Yonghyun-dong, Nam-gu, Incheon, 402-751, Korea
zhu@inhaian.net, kskwak@inha.ac.kr

Abstract. In this study, an adaptive handoff algorithm with a dynamic hysteresis value, based on the distance between the mobile station and the serving base station, is proposed for cellular communications. Handoff probability is calculated to evaluate handoff algorithms analytically. The proposed handoff algorithm is compared with an algorithm with fixed hysteresis, an algorithm using both threshold and hysteresis, and a handoff algorithm based on distance and hysteresis. Performance is evaluated in terms of the average number of handoffs, average handoff delay, standard deviation of handoff location, average signal strength during handoff, and probability of link degradation. Numerical results and simulations demonstrate that the proposed algorithm outperforms other handoff algorithms. The effect of distance error is also discussed.

1 Introduction

In cellular communications, handoff is the process of transferring the serving base station (BS) of a mobile station (MS) from one to another when the MS moves across a cell boundary. A call in progress could be forced to abort during handoff if sufficient resources cannot be allocated in the new wireless cell. A properly designed handoff algorithm is essential in reducing the switching load of the system while maintaining the quality of service (QoS). In this paper hard handoffs in a cellular system are concentrated on. The term handoff is henceforth used to refer to hard handoff.

When an MS is traveling from its serving BS to the target adjacent BS, the probability of handoff is generally designed to maximize at the cell boundary. The decision to initiate a handover may be based on different measurements [1]-[4] The received signal strength (RSS) measurement is one of the most common criteria. Traditional handoff algorithms depend on comparing the differential signal power level between the serving BS and target BSs to a fixed handoff hysteresis value h. This hysteresis value is designed to reduce the ping-pong effect in the handoff procedure. Therefore, selection of this hysteresis value becomes important for optimizing handoff performance. If h is too small, numerous unnecessary handoffs may be processed, increasing the network burden. However, if h is too large, the long handoff delay may result in a dropped-call or low QoS.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 122 – 131, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Two important performance indicators of a handoff algorithm are the average number of handoffs and the average handoff delay, both of which are required to be minimized. The size of the handoff area is a very important criterion relating to handoff. The handoff area should be small enough to avoid cell-penetration or cell-dragging. Cell-dragging occurs when excessive proportion of mobiles moves a considerable distance into neighboring cell areas without making a handoff, resulting in an increased level of system interference and a decrease in system throughput. The smaller the handoff area, the more performance improves. The standard deviation of the handoff location is an indicator of the size of the handoff area.

In order to improve handoff performance, various adaptive handoff algorithms were proposed [5]-[7]. MSs have recently been given the ability to continuously track the mobile location through various radio location techniques [8]-[10]. In this study, an adaptive handoff algorithm is developed by dynamically determining the hysteresis value as a function of the distance between the MS and the serving BS. Since the handoff hysteresis value is varied based on MS's location, it can intelligently reduce the probability of unnecessary handoffs and maintain the QoS.

The paper is organized as follows: First, the adaptive handoff algorithm is proposed in Section 2. Then, Section 3 analyzes the proposed handoff algorithm. Numerical results are presented to demonstrate the performance improvement compared with handoff algorithms with fixed hysteresis values. Performance evaluation is presented in Section 4 by comparing key performance criteria. The effect of distance error is discussed in Section 5. Finally, concluding remarks are presented in Section 6.

2 Proposed Handoff Algorithm

Many authors [11]-[15] analyzing handoff performance consider only two BSs and omit the effect of other adjacent BSs, which simplify performance analysis but result in inaccurate conclusions. In this study, the influence of all six adjacent cells is considered in the hexagonal cell model.

Since a mobile crossing a cell boundary experiences different propagation conditions based on the point at which it crosses the boundary, use of adaptive thresholds (as opposed to constant thresholds) is an attractive option for handoff algorithms. In this work, the distance between the MS and its serving BS is assumed to be known. A simple handoff algorithm with adaptive hysteresis value h, is proposed, which is determined by the distance between the MS and the serving BS, i.e.

$$h = \max\{20(1-(\frac{d_c}{R})^4), 0\}, \tag{1}$$

where d_c is the distance between the MS and serving BS, and R is the cell radius. Handoff is initiated only if the new BS's signal strength is sufficiently stronger by hysteresis value h, than that of the serving BS.

The coefficient is chosen to be 20, to enable the handoff algorithm to control unnecessary handoff and react to the deep fading simultaneously, resulting in sudden drop (20~30 dB) of RSS. The large exponent results in a decrease in the number of handoffs at the cost of increasing handoff delay and probability of link degradation. In

this paper, 4 is the default exponent for the proposed algorithm, because it is assumed that the probability of link degradation should not exceed 0.005.

As demonstrated in Equation (1), h decreases from 20 to 0 dB as the MS moves away from the serving BS. By setting the above dynamic hysteresis value h, the number of unnecessary handoff is decreased because of a large h if the MS is near the serving BS, and the MS is encouraged to hand over to adjacent cells because of a small h if it is near the boundary of the current cell. In this way, handoff area is optimized.

3 Analysis of Handoff Algorithm

3.1 System Model

It is assumed that the RSS is affected by path loss as well as the shadowing effect. In addition, Rayleigh fading exists. However, this is averaged out and can be ignored because handoff algorithms cannot respond to short-term fading [11]-[16]. Therefore, if the transmitted power of BS is normalized to be 0 dB, the signal strengths received from the current BS and adjacent BSs, denoted R_c and R_i, are given by

$$R_c = -K\log(d_c) + u(d_c), \tag{2}$$

$$R_i = -K\log(d_i) + v_i(d_i), i = 1,2,...,6, \tag{3}$$

where K represents the path loss factor, d_c and d_i represent the distance from the current BS and adjacent BSs respectively, and $u(d)$ and $v_i(d)$ model the effect of shadowing fading.

3.2 Handoff Analysis

Handoff occurs if one of the following conditions is satisfied:

$$[R_i > R_c + h], \quad i = 1,2,...,6, \tag{4}$$

where R_i and R_c are the RSS of the adjacent BSs and serving BS at the MS's location, respectively, and h is the hysteresis value. If the handoff condition is satisfied, the call is handed over to the adjacent cell with the largest R_i. Handoff will not occur unless the RSS from an adjacent BS is greater than that from the serving BS by the hysteresis value h. Therefore, the handoff probability P_{ho} is given as:

$$P_{ho} = P\{\bigcup_{i=1}^{6}[R_i > R_c + h]\}, \tag{5}$$

Remark 1. In order to compute handoff probability analytically at different locations in the cell, each location is treated independently. Therefore, the correlation distance d_{cor}, which determines the speed at which correlation of the shadowing process decays with distance when the link between MS and BS is considered at two time

instants (or at two different positions of the MS), is not included in the above equations. However, d_{cor} must be considered when a moving MS is considered for performance evaluation.

3.3 Numerical Results

Handoff probability for MS under the cover of the BS is computed at different locations. If signal power is noise free, the hysteresis value can be set to 0 such that the MS only performs handoff at the cell boundary; in this case, handoff probability is 1 at the cell boundary and 0 elsewhere. When noise is present, the handoff probability is zero near the center of the cell, and increases as the MS moves away from the serving BS till handoff occurs. A good handoff algorithm should have a handoff probability map approaching the ideal case.

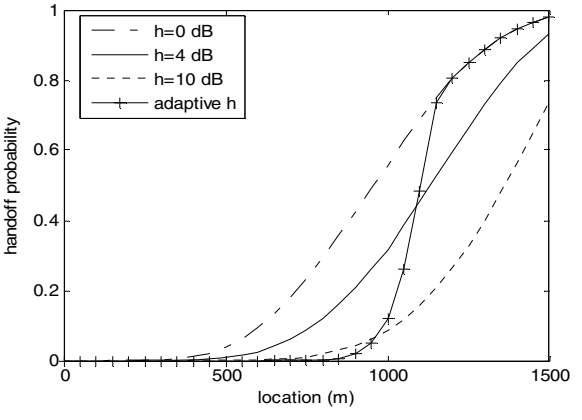


Fig. 1. Comparison of handoff probability for different *h*

Assuming the distance between two neighboring BS is 2000m, the handoff probability of the proposed algorithm when the MS moves away from the serving BS to an adjacent BS along a straight line, is compared with that of handoff algorithm with fixed h=0,4,10 dB, shown in Fig. 1. The algorithm demonstrates superior performance, because the handoff probability of the algorithm is smallest when the distance is less than 950m, and increases most rapidly from 0 to 1 in the handoff area near the cell boundary. Therefore, the proposed algorithm has the smallest handoff area.

3.4 Another Handoff Probability

In contrast to the handoff probability discussed above, for the MS locating in a cell geographically but communicating with the BS of another cell, another performance metric is the handoff probability from the serving BS to the BS of the cell in which the MS locates geographically, denoted by P_b. In Fig. 2, the MS locates in **Cell 0**, so its serving BS should be $\mathbf{BS_0}$, in order to balance the system traffic and reduce interference. However, its serving BS is $\mathbf{BS_1}$. In this case, P_b is the probability that the MS

hands over from **Cell 1** to **Cell 0**, i.e., from $\mathbf{BS_1}$ to $\mathbf{BS_0}$. For the ideal case, P_b is one within the cell area and zero elsewhere, if the signal power is noise free. Therefore, for an effective handoff algorithm, P_b should be one in most of the cell area and decrease rapidly in the area near the cell boundary.

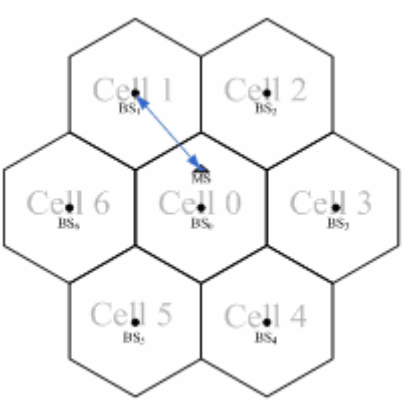


Fig. 2. Illustration of P_b

Assuming the serving BS is $\mathbf{BS_j}$, j=1,2,…6, handoff occurs from $\mathbf{BS_j}$ to $\mathbf{BS_0}$ if the following conditions are met:

1) If the RSS from $\mathbf{BS_0}$ exceeds that of $\mathbf{BS_i}$ for any $i \neq j$;
2) If the RSS from $\mathbf{BS_0}$ exceeds that of $\mathbf{BS_j}$ by hysteresis level h.

The corresponding handoff probability is

$$P_b^j = P\{\bigcap_{i=1, i\neq j}^{6} [R_c > R_i] \bigcap [R_c > R_j + h]\}. \tag{6}$$

Therefore,

$$P_b = \sum_{j=1}^{6} P_{BS_j} P_b^j \tag{7}$$

where P_{BSj} is the probability that the MS's serving BS is $\mathbf{BS_j}$, j=1,2,…,6.

Since P_{BSj} is extremely difficult to compute, if not impossible, the assumption that they have the same probability is taken, i.e., P_{BSj} =1/6, j=1,2,…,6. Fig. 3 illustrates the comparison of P_b between handoff algorithms with adaptive and fixed hysteresis when the MS moves away from the serving BS to an adjacent BS along a straight line. The proposed algorithm demonstrates superior performance, because the handoff probability of the proposed algorithm is the highest when the distance is less than 900m, and decreases most rapidly in the handoff area near the cell boundary.

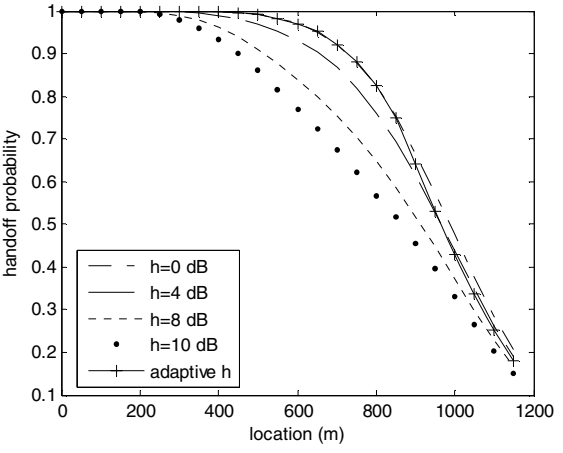


Fig. 3. Comparison of P_b

4 Performance Evaluation

The parameters used for simulation are presented in Table 1, which are commonly used to analyze handoff performance [11]-[15]. The MS is assumed to move from the serving BS to an adjacent BS along a straight line. The average number of handoffs, average handoff delay, standard deviation of handoff location, average signal strength during handoff and probability of link degradation are used as criteria for performance evaluation.

Table 1. Parameters used for simulations

Parameters	Description
D=2000 m	Distance between two BSs
R=D/sqrt(3)	Cell radius
K=30	Path loss factor
σ =5 dBm	Shadow fading standard deviation
d_{cor}=30 m	Correlation distance
v=20 m/s	Mobile velocity
t_s=0.5s	Sample time
Δ = -106 dBm	Threshold of link degradation

4.1 Comparison with Handoff Algorithm Using Both Threshold and Hysteresis

In regards to the handoff algorithm using both threshold and hysteresis, i.e., the handoff algorithm using both absolute and relative signal strength [11], the MS is handed over from one BS to another if both of the following conditions are met:

1) The RSS of the serving BS falls below an absolute threshold value T dB;
2) The RSS of the new BS becomes greater than that of the serving BS by a hysteresis of h dB.

Fig. 4 illustrates the comparison of handoff probability for handoff algorithms with adaptive hysteresis, fixed hysteresis, and both threshold and fixed hysteresis when the MS moves along a straight line from the serving BS to an adjacent BS. The handoff probability of the proposed algorithm is the smallest when the distance is less than 900m, and is the largest when the distance is greater than 1100m. The proposed algorithm is superior because it can control handoff when the MS is near the serving BS, and encourages handoff when the MS is leaving the serving cell.

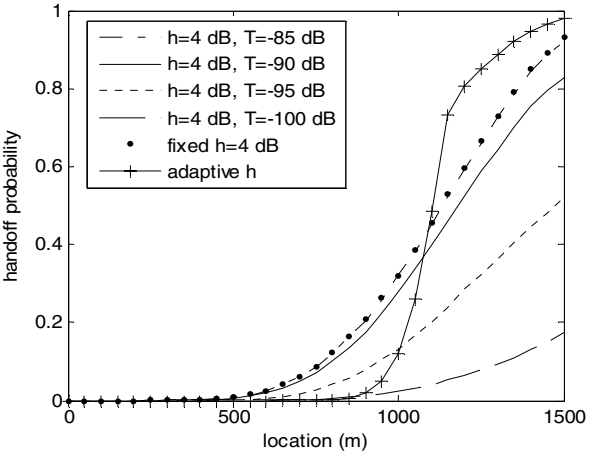


Fig. 4. Comparison of handoff probability

Table 2. Comparison of key criteria

Handoff Algorithm	Number of Hand-offs	Handoff Delay (m)	Standard Deviation (m)	Signal strength during handoff (dB)	Probability of Link Degradation
adaptive h	1.6521	44.902	85.607	-94.386	0.00458
Fixed h=4dB	8.2868	13.482	223.52	-93.298	0.00686
h=4, T=-80dB	8.3038	13.119	223.3	-93.301	0.00686
h=4, T=-85dB	8.2363	13.637	222.37	-93.415	0.0064
h=4, T=-90dB	7.1517	16.336	214.63	-94.416	0.00646
h=4, T=-95dB	3.7024	35.665	198.05	-97.34	0.00618
h=4, T=-99dB	1.9008	97.765	197.47	-99.834	0.01068
h=4, T=-100dB	1.549	144.54	205.68	-100.72	0.0153

A comparison of simulation results is presented in Table 3 The threshold has little effect on handoff performance if it is larger than -85dB. The average number of handoffs and signal strength during handoff decreases with decreasing threshold T, accompanied by an increase in handoff delay and probability of link degradation. In other words, the handoff algorithm using both absolute and relative measurements decreases the number of handoffs at the cost of increasing handoff delay, increasing probability of link degradation and decreasing signal strength during handoff.

As far as the average number of handoffs is concerned, the proposed algorithm can achieve similar performance as the handoff algorithm with h=4dB and T=-100dB. The

proposed adaptive handoff algorithm has smaller handoff delay and standard deviation of handoff area, which will decrease the interference on the neighboring cell and increase system throughput. Signal strength during handoff and the probability of link degradation of the proposed algorithm are much better. The improvements of the proposed algorithm are remarkable: 68.93% for handoff delay, 58.38% for standard deviation of handoff location, 6.334dB for average signal strength during handoff, and 70.07% for probability of link degradation.

4.2 Comparison with Handoff Algorithm Based on Distance and Hysteresis

As for the handoff algorithm based on distance and hysteresis [12], a handoff is performed to the adjacent base station if both the following conditions are met:

1) If the RSS from the adjacent BS exceeds that of the serving BS by a hysteresis level h dB;
2) If the measured distance from the serving base station exceeds that of the adjacent station by a threshold distance γm.

Table 4 demonstrates that this handoff algorithm can effectively decrease the number of handoffs by using a large distance threshold. The average number of handoffs decreases with increasing distance threshold at the cost of large handoff delay and high probability of link degradation.

Table 3. Comparison of key criteria

Handoff Algorithm	Number of Handoffs	Handoff Delay (m)	Standard Deviation (m)	Signal strength during handoff (dB)	Probability of Link Degradation
adaptive h	1.6521	44.902	85.607	-94.386	0.00458
Fixed h=4dB	8.2868	13.482	223.52	-93.298	0.00686
h=4, r=-0m	0.99216	55.916	52.111	-93.503	0.00972
h=4, r=100m	0.98776	95.61	44.176	-93.639	0.01348
h=4, r=200m	0.98364	137.47	37.518	-93.751	0.01696

The probability of link degradation is extremely high for the handoff algorithm based on distance and hysteresis, because the handoff algorithm cannot react to a sudden drop in signal strength when the MS is in the area where the distance condition of handoff is not satisfied. The call is forced to interrupt even though there is another feasible link, because handoff is not performed. This is also the reason why the standard deviation of handoff location is extremely small. From the subscriber's point of view, the blocking of handoff is less desirable than the blocking of a new call. Therefore, the handoff algorithm based on distance and RSS is not feasible because of its high probability of link degradation when call-drop probability is considered.

5 Effect of Distance Error

In this section, the effect of distance error is considered in the proposed handoff algorithm. The measured distance is assumed to be Gaussian distributed, where the mean is the accurate distance and the standard deviation is *std* [12].

Table 5 presents the results for different levels of location accuracy. The average number of handoffs increases with *std*, while average handoff delay decreases as *std* increases. There is a little change for the standard deviation of handoff area, signal strength during handoff and probability of link degradation. It can be concluded that the adaptive handoff algorithm is stable when the standard deviation of the measured distance is less than 60m, meaning that the proposed algorithm is still feasible when only rough distance information rather than accurate distance information is available.

Table 4. Effect of Distance Error

std (m)	Number of Handoffs	Handoff Delay (m)	Standard Deviation (m)	Signal strength during handoff (dB)	Probability of Link Degradation
0	1.6521	44.902	85.607	-94.386	0.00458
10	1.6562	44.505	85.769	-94.412	0.0046
20	1.6827	43.175	86.358	-94.327	0.00506
40	1.787	40.48	86.42	-94.188	0.0053
60	1.9893	36.061	87.761	-93.966	0.00468
80	2.2772	30.8	90.237	-93.665	0.0051
100	2.678	27.104	94.27	-93.389	0.00534
150	3.9001	19.961	111.37	-92.901	0.00442

6 Conclusion

In this study, an adaptive handoff algorithm is developed by dynamically determining the hysteresis value as a function of the distance between the MS and the serving BS. Since the handoff hysteresis value varies, depending on MS's location, it can intelligently reduce the probability of unnecessary handoffs and maintain quality of service. Analytical and simulation results demonstrate that the proposed algorithm performs better than other algorithms with fixed hysteresis value. And the algorithm is better than the handoff algorithm using both threshold and hysteresis and the handoff algorithm based on distance and hysteresis. It is also demonstrated that the proposed algorithm is stable when location error exists, i.e., it is location-error-resistant. When distance information is known, the only overhead of the proposed handoff algorithm is to compute the hysteresis value according to Equation (1), which is extremely simple. The proposed handoff algorithm is extremely effective and easy to implement when provided with distance information.

Acknowledgment

This research was supported by University IT Research Center Project of Inha UWB-ITRC, Korea. The work of Huamin Zhu was supported in part by Korea Science and Engineering Foundation (KOSEF).

References

1. Tripathi, N.D., Reed, J.H., VanLandinoham, H.F.: Handoff in cellular systems, IEEE Wireless Commun.unications, Vol. 5, No. 6 (1998) 26-37B
2. Pollini, G.P.: Trends in handover design, IEEE Communications Magazine, Vol. 34, No. 3 (1996) 82–90
3. Tekinay, S. and Jabbari, B.: Handover and channel assignment in mobile cellular networks, IEEE Commun. Mag. (1991) 42-46
4. Zonoozi, M., Dassanayake, P., Faulkner, M.: Optimum hysteresis level, signal averaging time and handover delay, Proc. IEEE 47th Veh. Technol. Conf., Vol. 1 (1997) 310-313
5. Vijayan, R. and Holtzman, JM.: A model for analyzing handoff algorithms, IEEE Trans. on Vehic. Techn., Vol. 42, No. 3 (1993) 351-356
6. Wang, S.S, Green, M. and Malkawi, M.: Adaptive handoff method using mobile location information, Broadband Communications for the Internet Era Symposium digest, 2001 IEEE Emerging Technologies Symposium on (2001) 97-101
7. Lau, S.S.-F., Cheung, KF, Chuang, J.C.I: Fuzzy logic adaptive handoff algorithm, Global Telecommunications Conference, 1995. GLOBECOM '95, IEEE, Vol. 1 (1995) 509-513
8. Fang, B.T.: Simple solutions for hyerbolic and related position fixes, IEEE Trans. Aerosp. Electron. Syst., Vol. 26 (1990) 748-753
9. Wallehnhof, H., Lichtenegger, H., and Collins, J.: Global Positioning System: Theory and Practice. Springer-Verlag, New York (1997)
10. Bajaj, R., Ranaweera, S.L., and Agrawal, D.P.: GPS: Location-Tracking technology, Computer, Vol. 35 (2002) 92–94
11. Zhang, N. and Holtzman, JM.: Analysis of handoff algorithms using both absolute and relative measurements, IEEE Trans. on Vehic. Techn., Vol. 45, No. 1 (1996) 174–179
12. Itoh, KI, Watanabe, S., Shih, J.S., and Sato, T.: Performance of handoff algorithm based on distance and RSSI measurements, IEEE Trans. on Vehic. Techn., Vol. 51, No. 6 (2002) 1460-1468
13. Prakash, R. and Veeravalli, V.V.: Adaptive hard handoff algorithms, IEEE J. Select. Areas Commun., Vol. 13, No. 11 (2000) 2456-2464
14. Veeravalli, V.V. and Kelly, O.E.: A locally optimal handoff algorithm for cellular communications, IEEE Trans. on Vehic. Techn., Vol. 46, No. 3 (1997) 351-356
15. Prakash, R. and Veeravalli, V.V.: Locally optimal soft handoff algorithms, IEEE Trans. on Vehic. Techn., Vol. 52, No. 2 (2003) 347-356
16. Gudmundson, M.: Correlation model for shadow fading in mobile radio systems, Electron. Lett, Vol. 27, No. 23 (1991) 2145-2146

A Seamless Service Management with Context-Aware Handoff Scheme in Ubiquitous Computing Environment

Tae-Hoon Kang, Chung-Pyo Hong, Won-Joo Jang, and Shin-Dug Kim

Dept. of Computer Science, Yonsei University, Seoul 120-749, Rep. of Korea
Tel.: +82-2-2123-2718; Fax: +82-2-365-2579
{thkang, hulkboy, loopng, sdkim}@yonsei.ac.kr

Abstract. Despite the importance of seamless connectivity in ubiquitous computing, research for seamless connectivity has not been considered properly. To provide seamless connectivity, this paper proposes an application execution model for seamless services with an intelligent context-aware handoff scheme based on context information, such as user intentions and application requirements as well as issues of network layer. In this paper, we define several profiles reflecting various context informations and design basic modules organizing middleware architecture for seamless service management and efficient vertical handoff (VHO) management. Consequently, the proposed seamless service management with context-aware handoff scheme provides service continuity in ubiquitous computing environment by preventing unnecessary handoffs, changing application service modes adaptively, and eventually maintaining optimal network configuration. And also simulation results show about 38% enhanced performance in the processing rate, about 13% enhanced performance from service continuity, and about 29% enhanced performance in the processing time, compared with conventional schemes.

1 Introduction

Recently, ubiquitous computing is regarded as one of the most attractive research topics in computer science, and thus both academic and industrial sectors conduct research in many relevant areas. One of the major premises for ubiquitous computing is that all components should be connected to the network [1]. For this reason, seamless connectivity is an important research issue in ubiquitous computing. Also to provide seamless connectivity, a handoff management scheme between heterogeneous network interfaces, called VHO, considering different characteristics of each interface and various context informations is required.

However, existing approaches only assume that all components are connected with networks seamlessly, and do not consider how seamless connectivity is provided. Also in the aspect of seamless service, existing research only focuses on adaptation of application itself or resource reservation to support quality of service (QoS). And they only consider one network interface and pay no attention to VHO between heterogeneous network interfaces. Thus, if current network interface does not satisfy a given condition, they cannot keep performing services. At the same time, in VHO schemes, existing handoff operation is only focused on safe and reliable handoff, and it only

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 132 – 141, 2006.
© Springer-Verlag Berlin Heidelberg 2006

occurs based on network factors, such as signal strength and latency without considering any context information, e.g., user intentions.

To provide seamless connectivity, this paper proposes an effective application execution model to provide seamless service by changing application service mode adaptively and proposes an intelligent context-aware handoff management scheme based on the profiles reflecting various context informations in ubiquitous computing environment composed of heterogeneous network interfaces. We define our own profiles and then we design basic modules organizing a middleware architecture for seamless service management and efficient VHO management, especially an application agent that changes application's service mode to control service quality for seamless service. Consequently, proposed seamless service management with context-aware handoff scheme considering user intentions as well as the issues of network layer provides service continuity in ubiquitous computing environment by preventing unnecessary handoffs, changing application service modes adaptively, and eventually maintaining optimal network configuration. And also simulation results show about 38% enhanced performance in the processing rate, about 13% enhanced performance from service continuity, and about 29% enhanced performance in the processing time, compared with conventional schemes.

The remaining part of this paper consists of four parts. In Section 2, related work is introduced. In Section 3, proposed schemes, middleware architecture, and algorithms, are presented. Section 4 shows evaluation result and, lastly, conclusions are presented in Section 5.

2 Related Work

In this section, precious research about ubiquitous computing, handoff schemes and seamless service will be described.

Pervasive computing infrastructure, GAIA, allows applications to specify different behaviors in different contexts by gathering of context information from different sensors, delivering appropriate context information to the applications and providing a powerful context model that allows complex reasoning to be done on contexts [2]. An architectural framework, AURA, allows user mobility and adapts computing environments proactively to a given situation by using models of tasks consisted of inferred service information and user information [3]. Context Toolkit [4] presents conceptual model for context-aware applications and enables the construction of prototypes for context-aware applications. Another pervasive computing infrastructure that exploits Semantic Web technologies to support explicit representation, expressive querying, and flexible reasoning of contexts in smart spaces, Semantic Space, enables applications to run proactively [5]. But, these approaches are about techniques for ubiquitous computing, like sensor abstraction, context reasoning, and so on, based on the assumption that seamless connectivity is guaranteed. They did not consider any interaction mechanism between changing environment and network status [2] [3] [4] [5].

In [6], they propose an adaptive QoS design approach via resource reservation and rate adaptation to support multimedia over wireless cellular networks. In [7], they propose the optimization of the QoS offered by real-time multimedia adaptive

applications through machine learning algorithms. These applications are able to adapt in real time their internal settings (i.e., video sizes, audio and video codecs, among others) to the unpredictably changing capacity of the network and guarantee a good user-perceived QoS even when the network conditions are constantly changing. But, these approaches only assume one network interface and only focus on resource reservation to support QoS. And they do not consider providing seamless connectivity through VHO by configuring applications [6] [7].

In [8], active application oriented (AAO) scheme performs VHO, based on application requirements rather than the issues of network layer, so it provides more efficient VHO. In [9], they propose a context-based adaptive communication system for use in heterogeneous networks. By using context information, they provide flexible network and application control. But in [8] and [9], they assume that only one application runs on mobile terminal and it does not take into account context information like user intention. And also in [9], they do not consider service continuity by configuring application's service quality.

3 Context-Aware Handoff Management

In this section, we present a seamless service management with context-aware handoff scheme considering context information that can configure the optimal network environment for providing seamless service and seamless connectivity. In Section 3.1, middleware structure is explained for each module. In Section 3.2 and Section 3.3, modules and major algorithms of VHO decision manager and application agent are explained. And Section 3.4, overall operation flow is presented.

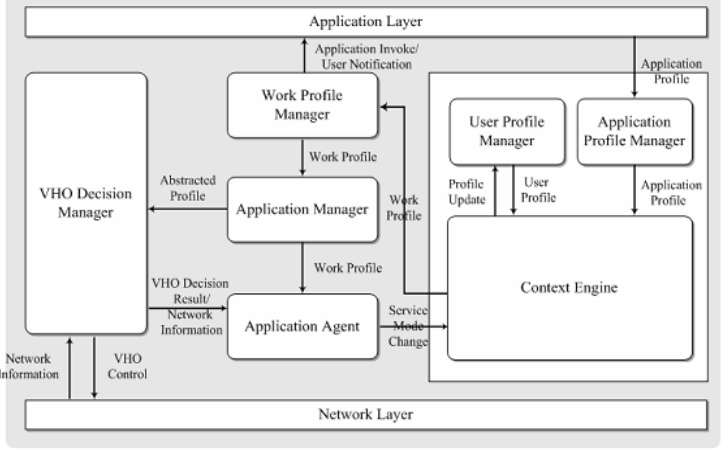


Fig. 1. Handoff management middleware architecture

3.1 Middleware Architecture

To support seamless service and seamless connectivity, optimal application and network configuration should be guaranteed. Thus, the proposed middleware architecture

consists of application profile manager, user profile manager, context engine, application manager, application agent, and VHO decision manager shown in Fig. 1.

Application profile manager manages application profiles reflecting some important application requirements. Application profile consists of several service modes that represent different service quality and minimum required values for bandwidth, packet error rate (PER), and latency in each service mode as shown as in Table 1.

Table 1. Example of application profile

Application Name	Service mode	Bandwidth (kbps)	PER (%)	Latency(ms)
Navigation	1 (video-based)	192	5	8
	2 (audio based)	64	8	12
	3 (text based)	16	10	15

User profile manager is to manage user profiles reflecting any specific user information and requirements. User profile is composed of using time, transferred data volume and priority in each work. Using time is the average time for a given work process completion in the previous work process and transferred data volume is the average amount of transferred data during the same process. Priority is the degree of preference for applications in each work. Example of user profile is shown as in Table 2.

Table 2. Example of user profile

Application Name	Work	Using Time (s)	Transferred Data Volume (Kbytes)	Priority
Navigation	Navigation	1920	8192	1

Context engine is an inference module based on application profile, user profile, and information from various sensors. In this paper, context engine is a simple rule-based module, and generates work profile. Work is a group of applications in some given situations, and work profile is the information related to the applications. Example of work profile is shown as in Table 3.

Table 3. Example of work profile

Work Name								
Navigation								
Application Name	Using Time (s)	Transferred Data Volume (Kbytes)	Priority	Current Service mode	Service Mode	Bandwidth (kbps)	PER (%)	Latency (ms)
Navigation	1920	8192	1	1	1(video-base)	192	5	10
					2(audio-base)	64	8	12
					3(text-base)	16	10	15

Work profile manager manages work profile, requests application invocation or the change of service mode, and notifies application suspension to the application layer.

Application manager generates abstract profile from the work profile. The abstract profile consists of bandwidth, packet error rate, latency, and work bandwidth. Bandwidth is the sum of all application's required bandwidths. Packet error rate and latency are determined by all application's requirements, which may be the minimum value among all application's values. Work bandwidth represents a permitted bandwidth based on user pattern and is used to prevent unnecessary handoff during VHO decision procedure. Its example is shown in Table 4.

Work bandwidth is calculated as follows:

$$\text{Work bandwidth} = \frac{\text{sum of all application's Transferred Data Volume}}{\text{Using Time of Work}}$$

Table 4. Example of abstract profile

Work Name	Bandwidth (kbps)	PER (%)	Latency (ms)	Work bandwidth(kbps)
Navigation	160	5	8	120

3.2 VHO Decision Manager

VHO decision manager decides whether it should perform any handoff and requests the network layer to change any chosen network interface if necessary, based on the abstract profile and network information. In this paper, we assume that network layer provides the information about other available network interfaces as well as current network interface through control channel. Detailed VHO decision algorithm is described as in Fig. 2.

```
check current network whether bandwidth, PER, latency of Abstract Profile are satisfied
if satisfied all factors then
   stay in current network
else
   check current network whether work bandwidth, PER, latency of Abstract Profile are satisfied
      if satisfied all factors then
         stay in current network
      else
         check available network whether bandwidth, PER, latency are satisfied
            if satisfied network exist then
               handoff to satisfied network
            else
               check available network whether work bandwidth, PER, latency are satisfied
                  if satisfied network exist then
                     handoff to satisfied network
                  else
                     handoff to nearest network to requirement
                     request Application Agent for Service mode change
```

Fig. 2. Pseudo code of VHO decision algorithm

3.3 Application Agent

Application agent changes service mode of any application to maintain service quality for seamless service. When VHO decision manager requests any service mode to be changed, application agent checks whether current network may satisfy work bandwidth or not. If current network is satisfied for work bandwidth, then enhance all

application's service quality and reactivate any suspended application if exists, and deliver this result to the context engine to update work profile. If current network does not satisfy work bandwidth, application agent temporally enhance all application's service quality and reactivate any suspended application if exists. Then application agent changes application's service mode based on each application's priority specified in work profile in the order of lower priority. If current situation is not improved by changing service mode, then the application agent suspends any application in the order of lower priority and delivers this result to the context engine to update work profile. Detailed application's service configuration procedure of application agent is described as in Fig. 3.

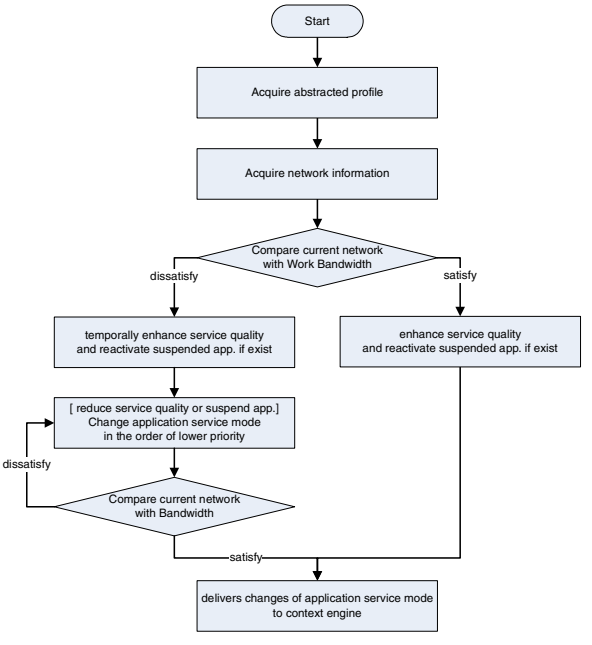


Fig. 3. Flowchart of application configuration procedure

3.4 Operation Flow

The overall operation progress based on the proposed scheme is performed as in the following steps. In the first step, context engine gathers user/application profile to generate its work profile and delivers it to the work profile manager. In the second step, work profile manager requests an application invocation to the application layer and delivers its work profile to the application manager. In this step, work profile manager requests to changes current service mode with suspension of any application to the application layer and notifies this result to user. In the third step, application manager generates abstract profile based on work profile and delivers it to the VHO decision manager. In the fourth step, VHO decision manager decides whether it should perform handoff and requests VHO to the network layer. And then application's service mode is

changed by the application agent if necessary based on algorithm described in Section 3.2. In the fifth step, application agent checks current application service mode and network interface information, and decides next service mode to change or any application suspension when VHO decision manager requests. Then the resulting status of any application is delivered to the context engine. And go to the first step for regenerating its associated work profile based on result of the fifth step.

4 Evaluation

In this section, we present simulation results to compare the proposed scheme with some other VHO schemes, conventional VHO scheme that only concerns network issues, and AAO (Active Application Oriented) scheme that concerns network issues and applications requirements.

4.1 Simulation Model

Simulation is performed with three network interfaces, seven applications, and four work groups. The three network interfaces are WWAN, DMB, and Wibro, and network factors about each interface are based on [10] [11] [12]. While simulation is proceeded, network factors of these network interfaces, bandwidth, PER, and latency, are randomly changed based on each interface's characteristics and the list of available network interfaces are randomly changed too. And theoretical specification of each network interface is shown as in Table 5. Seven applications are chosen as navigation service, broadcasting service, data transmission service, Instant Messaging Service (IMS), online game service, explorer service, and music service. Also four work groups are shown as in Table 6. In every simulation, applications for each work group have different using times and transferred data volumes, which are randomly generated.

Table 5. Example of network interfaces

Network Interface	Bandwidth	PER (%)	Latency (ms)	Coverage (Km)
WWAN	144 Kbps	3.5	30	1 ~ 10
Wibro	1 Mb	1.5	20	1
DMB	15 Mb	2.5	10	5 ~ 30

Table 6. Example of simulation scenarios

Work Groups	Applications
Entertainment	Broadcasting Service, IMS, Explorer Service
Online game	Online Game Service, Explorer Service
Business	IMS, Data Transmission Service, Explorer Service
Navigation	Navigation Service, Music Service

4.2 Simulation Result

Through the simulation result, three kinds of data can be obtained, i.e., the number of VHOs, the number of application failures, and the throughput. And based on these three kinds of data, we evaluate our proposed scheme compared with conventional scheme and AAO schemes in terms of three aspects, i.e., processing rate, service continuity and processing time.

Fig. 4 represents the processing rate, which is the percentage of overall throughput when ideal network environment exists. And according to the simulation result, the proposed scheme shows about 38% enhanced performance compared with conventional scheme and AAO scheme.

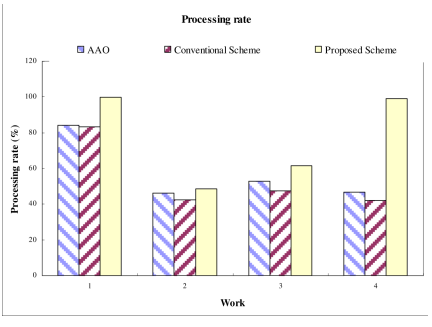


Fig. 4. Processing rate

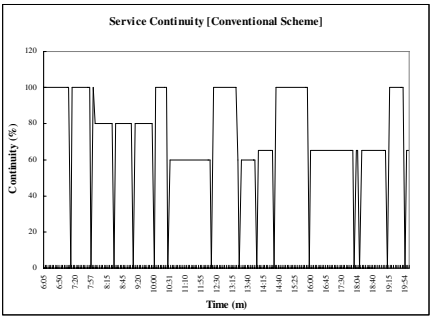


Fig. 5. Service continuity of conventional scheme

Fig. 5, 6, 7 represents the service continuity, which shows the amount of continuous computation calculated based on VHO delay and application's status reflecting any application suspension, namely application failure. In other words, service continuity represents how seamless service is maintained. And according to the simulation result, the proposed scheme in Fig. 7 shows about 13% enhanced performance compared with conventional scheme in Fig. 5 and AAO scheme in Fig. 6.

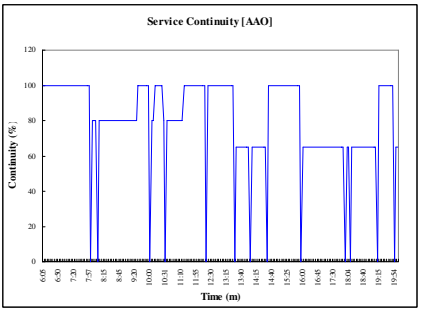


Fig. 6. Service continuity of AAO scheme

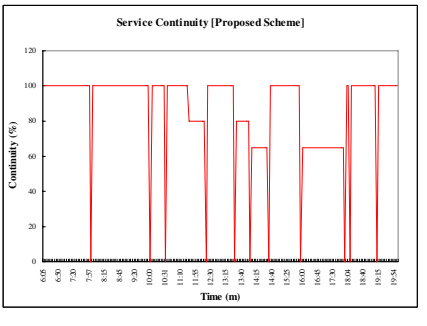


Fig. 7. Service continuity of proposed scheme

Fig. 8 represents the processing time, which is the time required for processing any specific amount of task. In Fig. 8, processing time of work group-1 shows the time spent for processing 450Mb task, processing time of work group-2 is the time spent for processing 600Mb task, processing time of work group-3 is the time spent for processing 450Mb task, processing time of work group-4 is the time spent for processing 450Mb task. And according to the simulation result, the proposed scheme shows about 29% enhanced performance compared with conventional scheme and AAO schemes.

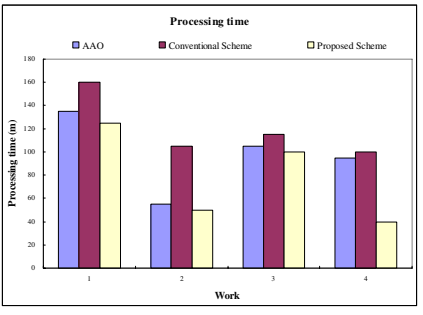


Fig. 8. Processing time

According to these simulation results, we can verify the effectiveness of proposed schemes, especially the algorithm of application agent. Application agent contributes about 70% of whole performance improvement by configuring service quality adaptively and by optimizing repercussion on VHO decision manager which reduces unnecessary handoffs.

5 Conclusion

Seamless connectivity is an important research issue in ubiquitous computing. However, research for seamless connectivity has not been considered properly and existing approaches only assume that all components are connected with networks seamlessly or are only concerned about factors of network issues. The provision of seamless connectivity requires an intelligent handoff management which builds up the optimal network environment and an application execution model which guarantees seamless service by configuring service quality of applications adaptively.

In this paper, we propose a context-aware handoff management scheme for seamless connectivity and a seamless service management scheme for continuous service in ubiquitous computing environment, based on profiles reflecting various context informations, such as user intentions as well as the issues of network layer. Consequently, the proposed schemes prevent unnecessary handoff operations, provide service continuity, and eventually enable an optimal network configuration in ubiquitous computing environment. And also simulation results show about 38% enhanced performance in the processing rate, about 13% enhanced performance from service continuity, and about 29% enhanced performance in the processing time, compared with conventional schemes.

References

1. Weiser, M.: The computer for the 21st century. Scientific American, vol. 265 (3), September 1991, page(s) 94-104.
2. Ranganathan, A. and Campbell, R. H.: An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Computing, vol. 7 (6), 2003, page(s) 353-364.
3. Sousa, J. and Garlan, D.: Aura: an architectural framework for user mobility in ubiquitous computing environments. Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture, August 2002, page(s) 29-43.
4. Dey, A.K., Salber, D. and Abowd, G.D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction (HCI) Journal, Vol. 16 (2-4), 2001, page(s) 97-166.
5. Wang, X., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., and Zhang, D.: Semantic Space: an infrastructure for smart spaces. IEEE Pervasive Computing, Vol. 3 (3), July-September 2004, page(s) 32-39.
6. Lu, S.W., Lee, K.W. and Bharghavan, V.: Adaptive Quality of Service Support for Packet-Switched Wireless Cellular Networks. Multimedia Tools and Applications, Vol. 17 (2-3), July 2002, page(s) 157–179.
7. Ruiz, P.M., Botia, J.A. and Gomez-Skarmeta, A.,: Providing QoS through machine-learning-driven adaptive multimedia applications. IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 34 (3), June 2004, page(s) 1398-1411.
8. Chen, W.T. and Shu, Y.Y.: Active Application Oriented Vertical Handoff in next-Generation Wireless Networks. IEEE Wireless and Communications and Networking Conference 2005, Vol. 3(13-17), March 2005, page(s) 1383-1388
9. Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M. and Morikawa, H.: Context-Based Network and Application Management on Seamless Networking Platform. Wireless Personal Communications, Vol. 35 (1-2), October 2005, page(s) 53-70.
10. Telecommunications Technology Association (TTA), http://www.tta.or.kr/
11. GSM World, GPRS-Standard class 10, http://www.gsmworld.com/
12. Media Independent Handover (MIH) Working Group, http://www.ieee802.org/21/

Performance Analysis of an Adaptive Soft Handoff Algorithm for Mobile Cellular Systems

Huamin Zhu and Kyungsup Kwak

Graduate School of Information Technology and Telecommunications
Room 428, High-Tech Center, Inha University
#253 Yonghyun-dong, Nam-gu, Incheon, 402-751, Korea
zhu@inhaian.net, kskwak@inha.ac.kr

Abstract. In this paper, an adaptive soft handoff algorithm, which dynamically calculates the threshold values for soft handoff based on the received signal strength, is proposed for mobile cellular communication systems. An analytical model is developed to study the performance of soft handoff algorithms. Performance is evaluated in terms of the average number active set updates, the mean size of the active set, the mean time of soft handoff, the average signal quality and link degradation probability, where a link degradation is defined to be the event that the signal strength falls below a level required for satisfactory communication. The adaptive soft handoff is shown to yield a significantly better tradeoff than the handoff algorithm with static thresholds.

1 Introduction

Handoff is an essential component of mobile cellular communication systems [1] [2]. It is the process whereby a mobile station (MS) communicating with one base station (BS) is switched to another BS when the MS moves across a cell boundary during a call. A call in progress could be forced to abort during handoff if sufficient resources cannot be allocated in the new wireless cell. A properly designed handoff algorithm is essential in reducing the switching load of the system while maintaining the quality of service (QoS). The design of reliable handoff algorithms is crucial to the operation of a cellular communication system and is especially important in microcellular systems, where the MS may traverse several cells during a call. The decision to initiate a handoff may be based on different measurements, such as the received signal strength (RSS) from the serving BS and neighboring BSs, the distance between the MS and the surrounding BSs, signal to noise ration (SNR), and bit error rate (BER). The RSS measurement is one of the most common criteria.

There are two types of handoff: hard handoff and soft handoff. Hard handoff is a break-before-make method, where a new link is set up after the release of the old link. A certain amount of margin is introduced to eliminate the ping-pong effect, which is the scenario of repeated handoff between two adjacent BSs caused by rapid fluctuations in the RSS from both of the BSs. Soft handoff is a make-before-break method [3]-[6]. With soft handoff, an active set is maintained, which is the set of all the BSs with which an MS is communicating. Depending on the changes in RSS from the two

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 142 – 151, 2006.
© Springer-Verlag Berlin Heidelberg 2006

or more BSs involved, a hard decision will eventually be made to communicate with only one BS. This normally happens after it is clear that the signal from one BS is considerably stronger than those from the others. In the interim period, the MS has simultaneous communication with all BSs in the active set. Generally, three parameters are specified in a soft handoff algorithm: the add threshold T_{add}, the drop threshold T_{drop}, and the drop timer D_{time}. When the pilot signal from a new BS exceeds that from the serving BS by a threshold value T_{add}, a new link to the new BS is established while maintaining the existing link. In this case, the call is said to be in soft handoff. We here assume that an MS can be in soft handoff with two strong BSs. When the RSS from either the old BS or the new BS weakens below T_{drop} and remains there for D_{time}, the bad connection is released and only a single good connection is maintained. The MS should reset and disable the timer if the level of RSS goes above the drop threshold T_{drop} before the timer expires.

The rest of this paper is structured as follows. The proposed adaptive soft handoff algorithm is presented in Section 2. Performance metrics to measure the performance of soft handoff are derived in Section 3. In Section 4, the simulation environment is described and the proposed adaptive handoff algorithm is compared with the static soft handoff algorithm. Finally, concluding remarks are presented in Section 5.

2 Proposed Adaptive Soft Handoff Algorithm

When an MS is traveling from its serving BS to a target BS, the probability of soft handoff is generally designed to maximize at the cell boundary. Traditional handoff decision algorithms compare the RSS to static or fixed handoff thresholds T_{add} and T_{drop} [3] [4]. Low thresholds and long drop timer settings ensure that there are a higher number of BSs in the active set and the call quality also tends to improve because of the higher diversity gain available. On the other hand, high thresholds and short drop timer settings increase the rate at which the active set updates but it saves network resources by maintaining a smaller number of BSs in the active set. However, the active set update requires signaling between the MS and BSs, which can be counted as another kind of system load. Therefore, selection of these handoff thresholds becomes very important for improving soft handoff performance.

The primary objective of a soft handoff algorithm is to provide good signal quality. Signal quality can be improved by including more BSs in the active set, but this comes at the cost of increased use of system resources. To lower the active set size, one option is to frequently update the active set so that the smallest active set with sufficient signal quality is maintained at each time instant. However, frequent updates or handoffs cause increasing switching overhead. Thus, a tradeoff exists among the following three metrics: the average number of active set updates, the mean size of the active set, and the average signal quality.

The algorithm presented here attempts to reduce the average number of active set updates and the mean size of the active set by dynamically adjusting the soft handoff thresholds based on RSS. Generally, high RSS means good signal quality, so another link between the MS and another BS can be considered unnecessary, because the MS is being served well by the current serving BS. The main idea in this paper is using high T_{add} to restrain other BSs from entering the active set and using high T_{drop} to

encourage other BSs in the active set to leave the active set when the RSS from a serving BS is much higher than the link degradation threshold.

A simple adaptive soft handoff algorithm with dynamic threshold values is proposed, which is determined by the RSS from the primary BS in the active set, i.e.

$$T_{add} = \min\{s_p - \Delta - 5,\ 10\}, \tag{1}$$

$$T_{drop} = T_{add} - 3, \tag{2}$$

where the primary BS is the serving BS with the largest RSS in the active set, s_p is the RSS from the primary BS, $\triangle$ is link degradation threshold.

Therefore, the following equation can be derived:

$$T_{add} = \begin{cases} 10, & s_p > \Delta + 15 \\ s_p - \Delta - 15, & s_p \le \Delta + 15 \end{cases}. \tag{3}$$

As demonstrated above, T_{add} equals 10 dB for s_p larger than $\triangle$+15, and it decreases from 10 dB to -5 dB as the RSS from the primary BS decreasing from $\triangle$+15 to $\triangle$. Number 15 is chosen because the link degradation probability at next time instant, denoted P_{ld}, is considerably low, if the MS is served by the same BS only. In this case, P_{ld} is approximately equal to $Q(15/\sigma)$, where $Q(\ )$ is the Q-function (complementary cumulative distribution function), σ is the standard variance of the shadow fading process. For instance, the link degradation probability is about 0.0062 for σ=6.

Since the handoff thresholds are varied based on RSS, it can intelligently reduce the probability of unnecessary handoffs while maintaining the signal quality. The handoff performance criterion is based on minimizing both the average number of active set updates and the mean size of the active set for the given propagation parameters and mobile path. The tradeoff between these two conflicting criteria is examined by comparing the proposed algorithm with a handoff algorithm with static thresholds in Section 4.

3 Performance Metrics

For the sake of simplicity, a basic system consisting of two BSs separated by a distance of D is considered in this paper [2]-[5]. And it is assumed that the MS is moving along a straight line with a constant velocity v between the two BSs, labeled $\boldsymbol{BS_1}$ and $\boldsymbol{BS_2}$. It is assumed that the RSS is affected by path loss as well as the shadowing effect. In addition, Rayleigh fading exists. However, this is averaged out and can be ignored because handoff algorithms cannot respond to short-term fading [3]-[6].

Let $d_i(n)$ denote the distance between the MS and $\boldsymbol{BS_i}$, i=1,2 at time instant n. Therefore, if the transmitted power of BS is normalized to be 0 dB, the signal strength from $\boldsymbol{BS_i}$, denoted $s_i(n)$, i=1,2, can be written as

$$s_i(n) = -K \log d_i(n) + u_i(n) \tag{4}$$

where K represents the path loss factor, and $u_i(n)$, i=1,2 are zero mean Gaussian random processes that model the log-normal shadow fading. The log-normal shadowing is assumed to have an exponential autocorrelation function [7] [8]

$$E[u_i(n)u_i(n+m)] = \sigma^2 a^{|m|} \tag{5}$$

where σ is the standard variance of the shadow fading process, and a is the correlation coefficient of the discrete-time fading process.

$$a = e^{-vt_s/d_c} \tag{6}$$

where t_s is the sampling time and d_c is the correlation distance determining how fast the correlation decays with distance.

The difference between $s_i(n)$ is defined as

$$x(n) = s_1(n) - s_2(n) \tag{7}$$

Let $P_{1\to12}(n)$, $P_{2\to12}(n)$, $P_{12\to1}(n)$, and $P_{12\to2}(n)$, be the probabilities that $\boldsymbol{BS_2}$ is added, $\boldsymbol{BS_1}$ is added, $\boldsymbol{BS_2}$ is dropped, and $\boldsymbol{BS_1}$ is dropped at time instant n, respectively. These probabilities can be computed as the following:

$$P_{1\to12}(n) = \Pr\left\{-x(n) > T_{add} \middle| -x(n-1) \le T_{add}\right\} \tag{8}$$

$$P_{2\to12}(n) = \Pr\left\{x(n) > T_{add} \middle| x(n-1) \le T_{add}\right\} \tag{9}$$

$$\begin{aligned} P_{12\to1}(n) = {} & \Pr\left\{-x(n-M) < T_{drop} \middle| -x(n-M-1) \ge T_{drop}\right\} \\ & \cdot \prod_{k=n-M+1}^{n} \Pr\left\{-x(k) < T_{drop} \middle| -x(k-1) < T_{drop}\right\} \end{aligned} \tag{10}$$

$$\begin{aligned} P_{12\to2}(n) = {} & \Pr\left\{x(n-M) < T_{drop} \middle| x(n-M-1) \ge T_{drop}\right\} \\ & \cdot \prod_{k=n-M+1}^{n} \Pr\left\{x(k) < T_{drop} \middle| x(k-1) < T_{drop}\right\} \end{aligned} \tag{11}$$

where

$$M = \lfloor D_{time} / t_s \rfloor \tag{12}$$

Let us define $P_1(n)$, $P_2(n)$, and $P_{12}(n)$ as the probabilities that the active set contains $\boldsymbol{BS_1}$ only, $\boldsymbol{BS_2}$ only, or both $\boldsymbol{BS_1}$ and $\boldsymbol{BS_2}$. Once the above transition probabilities are found, the assignment probabilities $P_1(n)$, $P_2(n)$, and $P_{12}(n)$ can be calculated under the initial condition $P_1(0)=1$, $P_2(0)=0$, $P_{12}(0)=0$.

Two performance measures, average number of BSs in the active set NO_{BS} (i.e., average size of the active set) and average number of active set updates NO_{update}, are given by

$$NO_{BS} = \frac{1}{N}\sum_{n=1}^{N}[P_1(n) + P_2(n) + 2P_{12}(n)] \tag{13}$$

$$NO_{update} = \sum_{n=1}^{N}\{P_1(n-1)P_{1\to 12}(n) + P_2(n-1)P_{2\to 12}(n) + P_{12}(n-1)[P_{12\to 1}(n) + P_{12\to 2}(n)]\} \tag{14}$$

However, the analytical results for the proposed adaptive soft handoff are intractable, because the thresholds T_{add} and T_{drop} are changing with the RSS. We will resort to simulation for performance evaluation in the next section.

4 Performance Evaluation

The parameters used for simulation are presented in Table 1, which are commonly used to analyze handoff performance [3]-[6]. The MS is assumed to move from the serving BS $\boldsymbol{BS_1}$ to an adjacent BS $\boldsymbol{BS_2}$ along a straight line. The average number of active set updates, the mean size of the active set, the mean time of soft handoff, the average signal quality and link degradation probability are used as criteria for performance evaluation. 50000 realizations were used to estimate the performance at each parameter setting in order to get stable results and smooth curves.

The tradeoff between the average number of active set updates and the mean size of the active set for soft handoff algorithms with different threshold values is illustrated in Fig. 1. T_{add} = -1dB and T_{drop} = -3dB are denoted as (-1,-3) and so on. It is clear that the proposed adaptive soft handoff algorithm has better performance than those with static thresholds. For soft handoff algorithm with static thresholds, the average number of active set updates NO_{update} decreases with the decrease of the drop threshold at the cost of increased mean size of active set NO_{BS} for the same add threshold; On the other hand, as the add threshold increases, both NO_{update} and NO_{BS} decrease for the same drop threshold. The cost accompanied with large add threshold is the increasing link degradation probability, shown in Fig. 2.

Table 1. Parameters used for simulations

Parameters	**Description**
D=2000 m	Distance between two BSs
R=D/sqrt(3)	Cell radius
K=30	Path loss factor
σ=8 dB	Shadow fading standard deviation
d_c=30 m	Correlation distance
v=20 m/s	Mobile velocity
t_s=0.5s	Sample time
M=3	Drop timer 1.5s
Δ= -105 dB	Threshold of link degradation

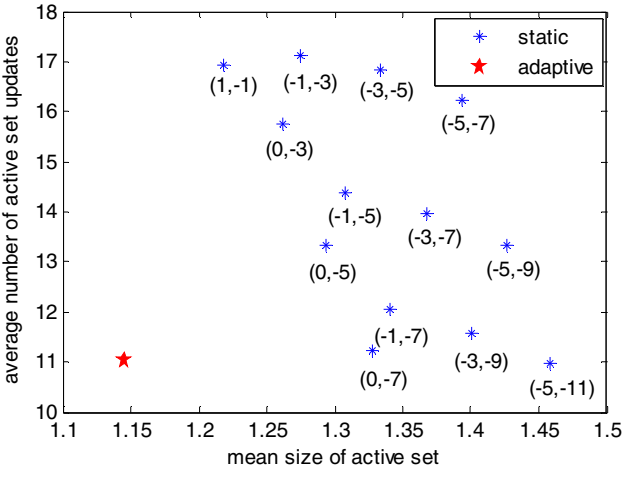


Fig. 1. Average number of active set updates versus average size of the active set

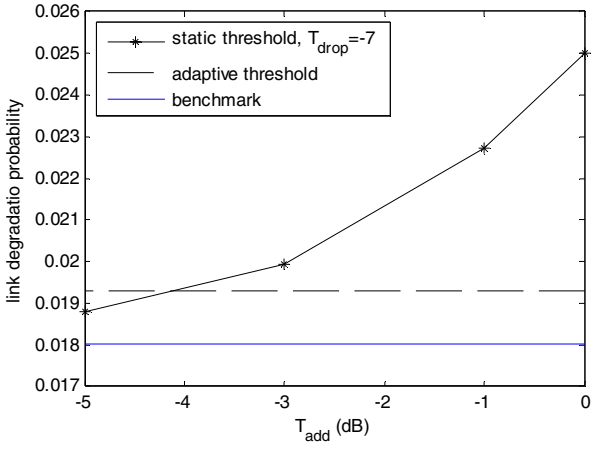


Fig. 2. Effect of T_{add} on link degradation probability

Fig. 2 shows the effect of the add threshold when the drop threshold is fixed, and Fig. 3 shows the effect of the drop threshold when the add threshold is fixed. The benchmark is also shown in the figures as the extreme case where the MS is communicating with both BSs at all time. The link degradation probability increases with the increase of either the add threshold or the drop threshold. The performance of the proposed adaptive handoff algorithm is approaching that of the benchmark, and is better than the handoff algorithm with static threshold values except the case with (-5, -7) whose link degradation probability is a litter lower. However, the average number of active set updates and mean size of active set of the handoff algorithm with (-5, -7) is much worse than those of the proposed adaptive handoff, shown in Fig. 1.

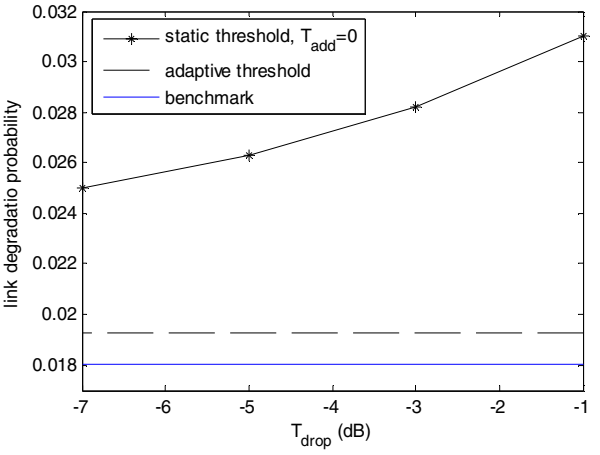


Fig. 3. Effect of T_{drop} on link degradation probability

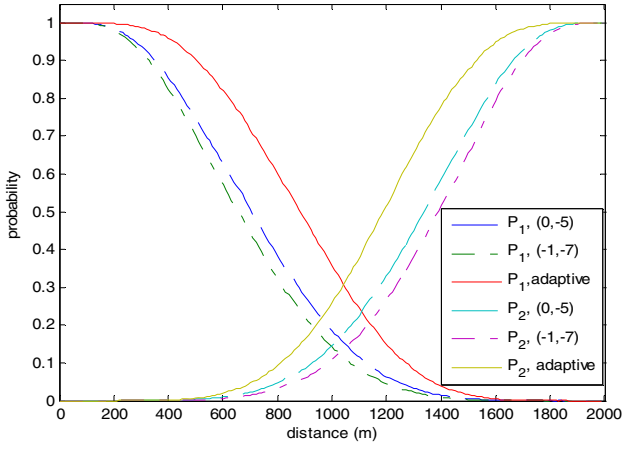


Fig. 4. Comparison of assignment probabilities

Fig. 4 shows the probabilities of assigning the MS to $\boldsymbol{BS_1}$ or $\boldsymbol{BS_2}$, and Fig. 5 shows the probabilities of soft handoff, i.e., the MS is assigned to both $\boldsymbol{BS_1}$ and $\boldsymbol{BS_2}$. The probability of soft handoff of the proposed adaptive algorithm is remarkably lower than those of static algorithms, so we can expect that the mean time of soft handoff of the proposed algorithm is shorter, shown in Table 2.

Comparison of link degradation probabilities at different location is shown in Fig. 6. The proposed adaptive algorithm shows the best performance. The tradeoff in decreasing the probability of soft handoff and enhancing link degradation

perfor-mance is a minor decrease in signal strength. Three call quality curves are compared in Fig. 7. The average signal strength decrease of the proposed algorithm is around 0.4dB.

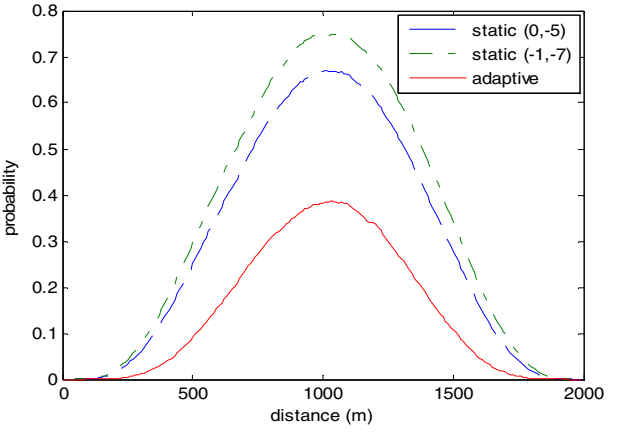


Fig. 5. Probability of soft handoff

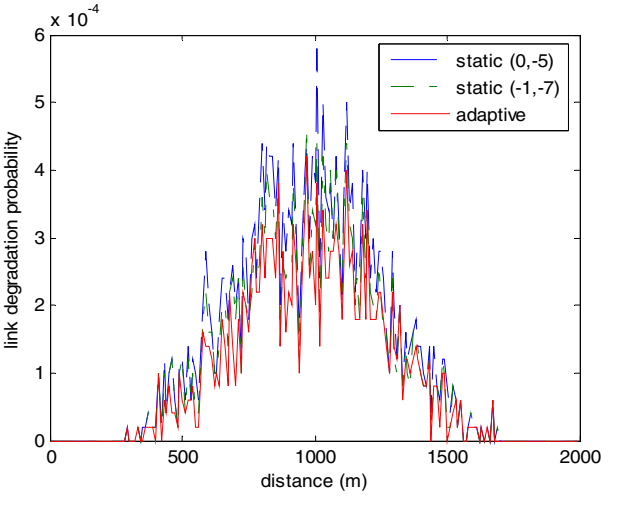


Fig. 6. Link degradation probability

To give a more quantitative comparison, the comparisons of key performance metrics, i.e., the average number active set updates, the mean size of the active set, the mean time of soft handoff, link degradation probability and the average signal quality, are presented in Table 2 for handoff algorithms with different parameters. We can conclude that the proposed adaptive algorithm achieves a better tradeoff among these performance metrics.

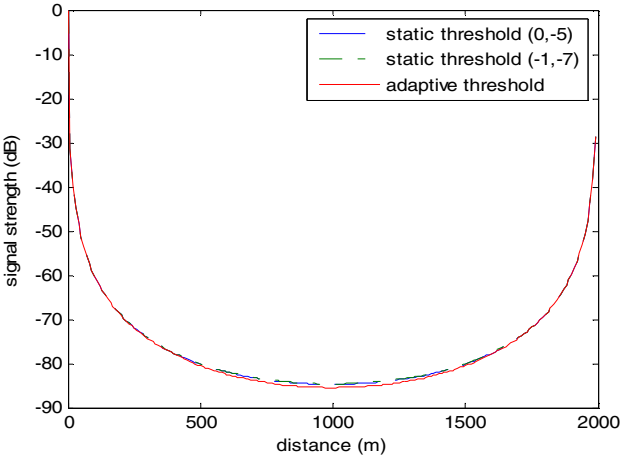


Fig. 7. Comparison of signal strength at different distance

Table 2. Comparisons of key performance metrics

Handoff algorithm (T_{add}, T_{drop})	Average size of active set	Average number of active set updates	Link degradation probability	Mean time of soft handoff (s)	Average signal strength
adaptive	1.145	11.02	0.0193	14.52	-75.83
(1, -1)	1.218	16.93	0.0353	21.81	-75.55
(0, -3)	1.261	15.74	0.0282	26.16	-75.49
(0, -5)	1.294	13.32	0.0263	29.40	-75.46
(0, -7)	1.327	11.22	0.0251	32.71	-75.44
(-1, -3)	1.274	17.12	0.0252	27.45	-75.46
(-1, -5)	1.307	14.39	0.0237	30.75	-75.44
(-1, -7)	1.341	12.06	0.0227	34.09	-75.42
(-3, -5)	1.334	16.85	0.0206	33.35	-75.41
(-3, -7)	1.367	13.96	0.0199	36.74	-75.39
(-3, -9)	1.401	11.58	0.0195	40.05	-75.37
(-5, -7)	1.393	16.22	0.0188	39.31	-75.36
(-5, -9)	1.426	13.33	0.0186	42.62	-75.35
(-5, -11)	1.458	10.97	0.0184	45.83	-75.34

5 Conclusion

An adaptive soft handoff algorithm with dynamic thresholds based on RSS is proposed in this study. The proposed adaptive soft handoff algorithm can achieve smaller average number of active set updates, reduce the mean size of the active set, and lower the link degradation probability at the cost of a minor decrease of the signal quality. Under the range of the parameters we have considered, the proposed adaptive handoff algorithm significantly outperforms handoff algorithms with static thresholds.

The only overhead of the proposed algorithm is calculating the thresholds according to Eq. (1) and (2), which is extremely simple.

Acknowledgment

This research was partly supported by University IT Research Center Project of Inha UWB-ITRC, Korea. The work of Huamin Zhu was supported in part by Korea Science and Engineering Foundation (KOSEF).

References

1. Tripathi, N.D., Reed, J.H., VanLandinoham, H.F.: Handoff in cellular systems. IEEE Wireless Commun.unications, Vol. 5, No. 6 (1998) 26-37
2. Wong, D., Teng Joon Lim: Soft handoffs in CDMA mobile systems. IEEE Wireless Communications, Vol. 4, No. 6 (1997) 6-17
3. Ning Zhang, Holtzman, J.M.: Analysis of a CDMA soft handoff algorithm. IEEE Transactions on Vehicular Technology, Vol. 47, No. 2 (1998) 710-714
4. Wang, S.S., Sridhar, S., Green, M.; Adaptive soft handoff method using mobile location information. IEEE 55th Vehicular Technology Conference, Vol. 4 (2002) 1936-1940
5. Akar, M., Mitra, U.: Soft handoff algorithms for CDMA cellular networks. IEEE Transactions on Wireless Communications, Vol. 2, No. 6 (2003) 1259-1274
6. Prakash, R., Veeravalli, V.V.: Locally optimal soft handoff algorithms. IEEE Transactions on Vehicular Technology, Vol. 52, No. 2 (2003) 347-356
7. Gudmundson, M.: Correlation model for shadow fading in mobile radio systems. Electronics Letters, Vol. 27, No. 23 (1991) 2145-2146
8. Graziosi, F., Santucci, F.: A general correlation model for shadow fading in mobile radio systems. IEEE Communications Letters, Vol. 6, No. 3 (2002) 102-104

An Admission Control and Traffic Engineering Model for Diffserv-MPLS Networks

Haci A. Mantar[1,2]

[1] Department of Computer Engineering, Gebze Institute of Technology, Turkey
[2] Department of Computer Engineering, Harran University, Turkey

Abstract. This paper presents a Bandwidth Broker (BB) based admission control and traffic engineering model for Diffserv supported MPLS networks. The proposed model uses a multi-path model in which several paths are pre-established between each ingress-egress router pair. As a central agent in each domain, the BB performs admission control on behalf of its entire domain via pre-established paths. The proposed model reduces the network congestion by adaptively balancing the load among multiple paths based on measurement of path utilization state. It increases the network core scalability by minimizing the core routers' state maintenance and signaling operation. The experimental results are provided to verify the achievements of our model.

1 Introduction

A significant research effort has been done to support Quality of Services (QoS) in the Internet. The research community has focused on three common architectures: Integrated Services (Intserv), Differentiated Services (Diffserv) [4] and MultiProtocol Label Switching [3]. While Intserv with RSVP signaling provide excellent QoS guarantees, it has scalability problems in the network core because of the per-flow state maintenance and the per-flow operation in routers. Because of scalability problem with this model, the IETF has proposed Diffserv [4] as an alternative QoS architecture for the *data/forwarding plane*. Diffserv does not have per-flow admission control or signaling and, consequently, routers do not maintain any per-flow state or operation. Core routers merely keep states for a small number of classes named Per Hop Behavior(PHB), each of which has particular scheduling and buffering mechanisms. A packet's PHB is identified with the Diffserv code point field (DSCP) assigned by the ingress router.

To this end, Diffserv is relatively scalable with large network sizes because the number of states in core routers are independent of the network size. Thus, it is considered as the *de facto* QoS standard for the next generation of the Internet. However, unlike the Intserv/RSVP, Diffserv only addresses data/forwarding plane functionality, whereas control plane functions still remain open issues. 1) A PHB defines the forwarding behavior in a single node. There is no QoS commitment for the traffic traversing multiple nodes; 2) With the exception of Expedited Forwarding (EF), all the PHBs provide *qualitative* QoS guarantees (the QoS metrics value changes with network conditions). Hence, the requirements of

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 152–161, 2006.
© Springer-Verlag Berlin Heidelberg 2006

real-time applications, which need *quantitative* bounds on specific QoS metrics, cannot be guaranteed even in a single node; 3) There is no admission control mechanism to ensure that the total incoming traffic to a node or domain does not exceed the resources.

MPLS specifies ways that Layer 3 traffic can be mapped to connection-oriented Layer 2 transport like ATM and Frame Relay. It adds label containing specific routing information to each packet and allows routers to assign explicit paths to various class of traffic. MPLS also provides network traffic engineering and simplifies data forwarding by establishing explicit paths, Label Switching Paths (LSPs), between ingress and egress router. However, similar to Diffserv, MPLS lacks of admission control. It does not address QoS guarantees. Furthermore, providing scalable QoS routing, which is an essential component of MPLS, is still an open issue.

As seen, neither Diffserv nor MPLS alone can provide QoS guarantees. However, the combination of both DiffServ and MPLS can be a base for providing QoS guarantees across a network. In this work we propose an admission control and traffic engineering framework for quantitative QoS guarantees for Diffserv-MPLS networks. We introduce a pre-established multi path model in which several LSPs with a certain bandwidth constraint are established for each ingress-egress pair in offline. The proposed model uses a Bandwidth Broker (BB) [1] as a control agent. The BB performs admission control based on the resource availability in LSPs and dynamically modifies the paths' capacities according to network conditions.

2 Network Model and Assumptions

We assume that a set of PHBs providing quantitative QoS guarantees are supported by each router [11]. A quantitative PHB i is associated with an upper delay bound d_i, an upper loss ratio bound l_i, and certain percentage of link capacity C_i (e.g., $d_i < 3ms$, $l_i < 0.01\%$). Each PHB can use only its share of link capacity, and the surplus capacity of a PHB can be used by best-effort or qualitative services. We also assume that d_i and l_i are pre-determined at the network configuration stage [8][9][11], which is done in relatively long time intervals (e.g., weeks) and downloaded into routers. A router dynamically adjusts its scheduler rate and buffer size according to the dynamic traffic rate to meet pre-determined d_i and l_i constraints. Under this premise, each router provides the desired QoS regardless of the utilization rate. (For details on quantitative PHB, the reader is referred to [8][9][11]).

To extend Diffserv QoS from a single node to across a domain, the IETF has introduced Per-Domain Behavior (PDB)[5]. A PDB defines the upper bounds of QoS constraints that identifiable packets will receive across a domain, regardless of the network utilization, from ingress router to egress router (IR-ER). In this sense, PDBs associates with QoS characteristics of LSPs.

Note that since the link resources allocated for a PHB can only be used by that PHB, the network can be considered as if it is divided into multiple virtual

networks, one for each PHB. Thus, in the rest of this paper, it is assumed that there is only one PHB within a network.

3 Admission Control and Traffic Engineering

3.1 Overview

Providing QoS guarantees across a network has at least three challenging problems. First, under dynamic and unpredictable traffic conditions it is difficult to maintain accurate network state information in a scalable manner. Because network resource availability may change with each flow arrival and departure, and therefore a high frequent update is required to maintain accurate network state. Second, performing QoS routing for each request can cause a serious scalability problem, because QoS routing is a computationally intensive task. Third, QoS routing may find a different path for each request. Since a path needs a state in routers, the number of path states may increase accordingly with the number of accepted requests.

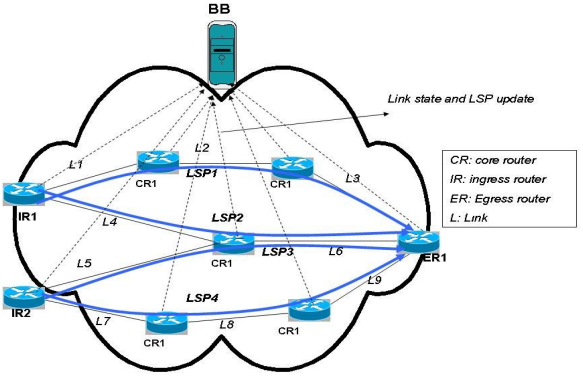


Fig. 1. A Diffserv-MPLS Resource Control Architecture

We propose a pre-established multi-path model that aims to minimize the above scalability problems while achieving efficient resource utilization and QoS guarantees. In general, our model has twofold: quantitative QoS guarantees and load balancing (traffic engineering) across a domain. Similar to [2][7], we assume that several explicit paths, MPLS paths (LSPs), have been pre-established between each ingress router (IR) and egress router (ER) with a certain bandwidth capacity (Fig.1.). The flows arriving at a IR are forwarded along one of these LSPs associated with IR-ER pair. Note that the multi-path routing model is not new, it has been proposed by several studies such as [2][7]. However, these studies have not addressed admission control, meaning that no mechanism to control incoming traffic. Therefore, they can not provide quantitative QoS guarantees. To be more specific, our model has the following features:

- Each LSP has a reserved bandwidth capacity
- The size of LSPs are dynamically adjusted with respect to network traffic conditions.
- LSPs are PHB-specific, meaning that all the traffic in an LSP belongs to the same PHB (an LSP is defined with *<IR, ER, PHB>*). This also means that all the LSPs associated with a *<IR, ER, PHB>* has the same PDB (QoS characteristics).
- The traffic entering an LSP is statistically guaranteed to be delivered to the other end of the LSP (egress router) within QoS boundaries specified by the associated PDB, regardless of the network congestion level.

For admission control and LSPs' resizing, we use a BB [1]. Upon receiving a reservation request, the proposed BB first determines the corresponding IR and ER. It then checks the total resource availability in the associated LSPs (with that particular IR-ER pair). If there is enough resource, it grants the requests, otherwise it rejects.

As depicted in Figure 1, each router within the domain sends its link state QoS information (for each interface-PHB pair) directly to the BB rather than flooding to all other routers as done in traditional link state protocol. The ingress routers also send the state of their LSPs (i.e., current traffic rate) to the BB. The BB dynamically checks the traffic rate of LSPs. When the traffic rate of an LSP reaches its maximum capacity, the BB attempts to resize the LSPs. The new sizes of LSPs are determined according to their current utilization rate. By resizing LSP with respect to their current utilization rate, overall network load is balanced, congestion is minimized and admission control probability is increased.

Once requests are accepted (by the BB), it is the role of ingress router to distribute the aggregated load across LSPs according to their perceived path costs, which reflects their utilization/congestion level. This is done in two phases, compute the cost of LSPs and balance the load across them. The routers along LSPs send their link cost to the associated ingress router periodically (e.g., every 5 sec). By having the cost of all the links along its LSPs, the ingress router computes the cost of each LSP by simply adding the cost of all the links constituting the LSP. It then attempts to equalize the cost of LSPs. The intuition here is that if utilization of an LSP i is greater than utilization of an LSP j, then the cost difference is minimized by shifting some load from i to j. This increases the cost of j and decreases the cost of i and equilibrium state is reached when they are equal. An important point here is that shifting does not violate QoS commitments (Sec. 3.3), because the shifting process is done among the LSPs that have the same QoS characteristics (the same QoS bound/PDB).

3.2 LSP Cost Computation and Update

Each link in a domain has the cost as one of its QoS attributes. The idea behind using the link and LSP cost is to shift the traffic from the congested path(s) to the less congested one(s). Thus, the cost of a path should reflect the path's

congestion. Among many possible cost functions that exhibit this condition, we choose a simple one. In this function a link cost for a PHB x is simply computed as

$$c_x(t) = q_x/(1 - u_x(t)) \tag{1}$$

where q_x is the fixed cost of using the link for x when it is idle, and $u_x(t)$ represents the link utilization of x at time t. $u_x(t) = r_x(t)/R_x$, where $r_x(t)$ is the traffic rate of x at time t and R_x represents the link capacity assigned to x.

The idea of the LSP cost update is very simple. Let $L1, L2, L3, ..LN$ be the links on an LSP i. The ingress router periodically receives the cost each of these links. Once the ingress router has the cost of all links, it computes the cost of i as

$$Ci = \sum_{j=1}^{N} c_j \tag{2}$$

3.3 Ingress Equal-Cost Load Balancing

Given the traffic load flowing between an ingress-egress pair, the task of an ingress router is to distribute the traffic across the candidate LSPs in a way the loads are balanced and therefore congestion is minimized. This is achieved through the cost-equalization of the LSPs. As described above, the ingress router periodically computes its LSPs' costs. When a persistent cost change is detected, the ingress router invokes the load balancing/cost-equalization algorithm that proportionally distributes the load across LSPs with respect to their perceived cost. The idea here is that upon detecting a consistent and substantial cost difference among LSPs, the ingress router shifts some of the traffic from the high-cost LSP(s) to the low-cost LSP(s). The cost equalization can be done using stochastic approximation theory or gradient projection methods, which are generally complex to implement in practice. To circumvent this problem, we use a simple but effective iterative procedure.

Let $C_1, C_2, ..., C_K$ be the costs, $R_1, R_2, ..., R_K$ the traffic rate and $\alpha_1, \alpha_2, ..., \alpha_K$ load proportions of $LSP_1, LSP_2, ..., LSP_K$ between an IR-ER pair and let V be total load of a IR-ER pair. If all C_i are equal, then $\alpha_i s$ are the desired proportions. If not, we use mean costs of all the paths $C^- = \sum C_i/K$ as the target cost for each path and obtain new proportions. The new proportions are computed as

$$\alpha_i^{'} = \frac{C^-}{C_i}\alpha_i \tag{3}$$

To normalize $\alpha_i^{'}$, we define a normalization factor $\phi = \frac{1}{\sum_{i=1}^{K} \alpha_i^{'}}$. The new values are obtained as $\alpha_i = \phi\alpha_i^{'}$. The corresponding R_i's are: $R_i = \alpha_i V$. This procedure is repeated iteratively until the costs of LSPs are equal. Because for a constant LSP size (during the load balancing) the cost C_i is increased with its load ($\alpha_i V$), it can be shown that this procedure will always converge.

For clarification, consider Figure 1, where there are four LSPs, LSP1, LSP2, LSP3, and LSP4. Let C1, C2, C3, and C4 be the costs and R1, R2, R3 and R4 be the traffic rate of LSP1, LSP2, LSP3 and LSP4, respectively. Suppose that the ingress routers IR1 and IR2 detect substantial cost differences between their LSPs, $C2 > C1$ and $C3 > C4$. In this situation, both IR1 and IR2 start shifting some traffic from LSP2 to LSP1, and from LSP3 to LSP4, respectively. That is, R1, R2, R3 and R4 will become R1+$\triangle r1$, R2-$\triangle r1$, R3-$\triangle r2$, and R4+$\triangle r2$, respectively. This process is repeated until there is no appreciable cost difference.

Another important issue is how to assign the load to the LSPs. We use a flow-based hashing approach in order to avoid out-of-order packet delivery. In this approach, hashing is applied on source and destination IP addresses and possibly other fields of the IP header. In the hashing model, the traffic will be first distributed into N bins by using module-N operation on the hash space [6]. The router performs a modulo-N hash over the packet header fields that identify a flow. If the total traffic rate is X bps, each bin approximately receives the amount of X/N bps. The next step is to map N bins to LSPs. The number of bins assigned to an LSP is determined based on its load portion.

3.4 Dynamic Update of LSP Capacity

As described before, in our model each LSP is assigned to a certain bandwidth amount. The capacity reserved for an LSP can not be used by others. Thus, some LSPs may have unused bandwidth while other LSPs rejecting requests due to the lack of bandwidth. In such cases, the LSPs needs to be resized in order to increase the network resource utilization. As depicted in Figure 1, when an LSP reaches its utilization target (e.g., 95% of its maximum capacity), the ingress router sends a pipe resizing message to the BB. Upon receiving a resizing message from any of ingress routers, the BB resizes some or all of the LSPs. The BB performs this task in two stages: It obtains the QoS state information of all the links along the path and then determines an appropriate path size.

The LSPs are resized based on their current utilization rate. Although the BB has the knowledge of all the links within the domain, this may not be sufficient to determine an LSP size because a link (its capacity) may be used by multiple paths. We therefore use the virtual capacity to define the amount of bandwidth that can be used by an LSP routed through that link.

Suppose m LSPs, $P1, P2...Pm$ share a bottleneck link k that has capacity c_k. Let P_i^{cl} denote the current traffic load of P_i that uses c_k. The virtual capacity, vc_{ik}, of P_i on link k can be obtained as

$$vc_{ik} = c_k \frac{P_i^{cl}}{\sum_j^m P_j^{cl}} \tag{4}$$

The vc is computed for each link using equation (4). Each LSP has its own share of a link as if it is the only one that uses that link. As seen in (4), when several LSPs compute for the same resources, the resources are assigned based on their utilization rate. After computing the vc of each link, the path size is determined as

$$P_i^{size} = min(vc_{i1}, vc_{i2}, ..., vc_{in}) \quad (5)$$

For clarification, consider the following scenario: The capacities of LSP1, LSP2, LSP3, LSP4 set to 6, 10, 10, and 6 Mbps, respectively (Fig.1). And the links L4, L5, L6 have 20Mbps capacity and all the other links have 10Mbps capacity. Let say after some time the loads of $IR1 - ER1$ and $IR1 - ER2$ become 15Mbps and 10Mbps, respectively. After IR1 and IR2 perform the load balancing, the traffic rate of LSP1, LSP2, LSP3 and LSP4 will approximately become 6, 9, 4, and 6Mbps. (As seen, $IR2$ assigns more traffic to LSP4 than LSP3 in order to minimize the cost difference between LSP3 and LSP4, in turn, minimize congestion and thus increase utilization.) Also as seen, the load of $IR1$ reaches its maximum utilization target (the traffic rate of LSP1 reaches its maximum capacity). In this case, $IR1$ will send a message to the BB asking to resize its LSPs' capacities. Upon receiving the resizing message, the BB will first compute the vc of each LSP in shared links and then set the size of LSPs based on equation 4 and 5, respectively.

This scheme has several appealing features: First, as long as the traffic rate in an LSP does not exceed its utilization target, the resizing process is not invoked. Second, all the traffic that has the same ingress-egress pair is aggregated, regardless of its source and final destination. This coarse level of aggregation damps the short-live traffic fluctuations, consequently reduces the frequency of LSP resizing process.

After the traffic rate of each LSP is computed and sent to the BB, the admission control becomes very simple. Upon receiving a request, the BB checks the total available capacity of corresponding LSPs. If the request's bandwidth amount is less than the total available bandwidth, it accepts the request otherwise it rejects.

4 Evaluation Results

In this section, we present experimental results to verify that the proposed model is stable, robust and scalable in such a way that it minimizes congestion and quickly balances the load cross multiple paths between IR-ER pairs in a reasonable period of time. We modified our previous software tool [11] for this purpose. We had a network topology consisting 8 edge routers (ingress, egress) and 8 core routers (CR) (Figure 2). Edge routers had mesh connections. For each IR-ER pair, two LSP were established, so there were 32 LSPs. Although an IR had LSP connection to all the ERs, in the figure the connection to a single ER is shown (for simplicity illustration).

Figure 3 and 4 shows, the load-balancing (cost-equalization) process for cost update interval of 5 seconds and 1 seconds. In this experiment, the offered average traffic rate between each IR-ER pair was changed every 60 sec. The figures shows utilization of L1, L2 and L3 under different traffic conditions. As figures shows, our model achieves load-balancing in few iterations.

Figure 5 shows the importance of our measurement-based paradigm in achieving the pre-determined QoS constraints. We verified QoS assurance only through

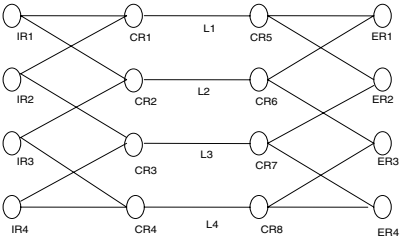


Fig. 2. The simulation topology

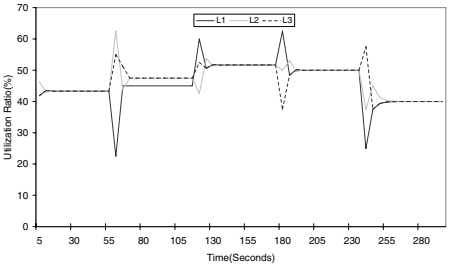


Fig. 3. The LSPs' load balancing

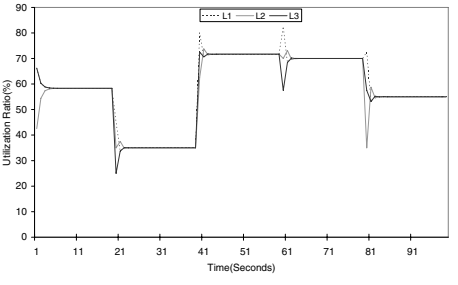


Fig. 4. The LSPs' load balancing

loss ratio rate. For measurement intervals of 10 seconds and tolerable loss ratio rate of 0.1%, the priority service met its given loss-ratio constraints with 99.92%. These QoS quantitative achievements come with the strict admission control mechanism.

Figure 6 shows the comparison of the pre-established path scheme with the shortest path and on demand path schemes. In on demand models such as [10], the paths are computed based on requests, and the sessions are not rerouted, meaning that there is no shifting process. As expected, the proposed model is much better than the shortest path scheme when the load is high. The interesting result here is that our scheme is even better than on demand scheme (it accepts more requests). This is because of the lack of traffic shifting with the on demand scheme. Note that the on demand path schemes also have serious scalability problems. Because they performs the QoS routing and path reservation set up for each individual requests.

As described before, upon receiving a reservation request, the BB performs the admission control based only on the current utilization of the corresponding LSPs. It does not check the capacity of the links constituting the LSPs. In on demand scheme (the traditional approach of a BB), the BB first determines the path and then check the capacity of all the links along the path. As shown in Figure 7 and 8, our approach significantly decreases the BB admission control time.

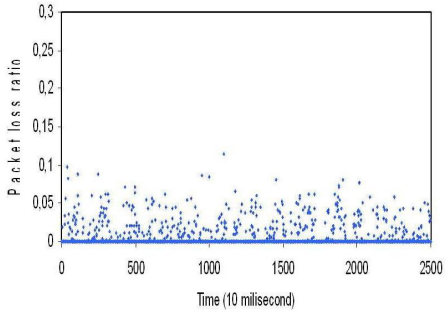


Fig. 5. Statistical QoS assurance

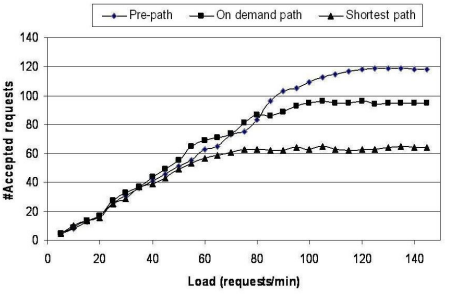


Fig. 6. Resource utilization

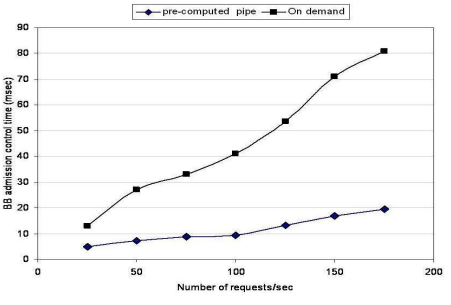


Fig. 7. The BB admission control time

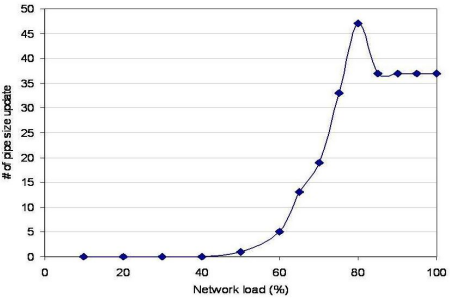


Fig. 8. The BB pipe resizing frequency

Figure 7 depicts the BB pipe resizing frequency under different network load. As shown, when the network is lightly-loaded, the resizing frequency is very small. However, when the network is heavily-loaded (e.g., more than 90%), the pipe resizing increases exponentially. This can result in serious scalability problem, because the resizing process is computationally intensive. To reduce this problem, the BB uses periodic update scheme for heavily-loaded network conditions. For example, when the pipe resizing frequency exceeds some predetermined threshold, the BB resizes the pipes periodically. As shown 8, the periodic update keeps the the resizing frequency in a reasonable range.

5 Conclusion and Future Work

In this paper we proposed a pre-established multi-path model, in which several MPLS paths (LSPs) with certain bandwidth constraint are pre-established for each ingress-egress pair. The proposed model performs admission control based on the resource availability of the paths without signaling the nodes along the paths. The loads among paths are balanced based on their utilization costs dynamically obtained via measurements. Our model significantly reduces admission

control time and minimizes scalability problems presented in prior research while optimizing network resource utilization.

In this work we focused only on the quantitative services, in the future, we will extend our model for the qualitative services. We will also study the theoretical verification of stability.

References

1. K. Nichols, V. Jacobson, and L. Zhang. " A Two-bit Differentiated Services Architecture for the Internet" RFC 2638, July 1999.
2. A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", IEEE INFOCOM 2001
3. E. Rosen, A. Viswanathan, R. Callon "Multiprotocol Label Switching Architecture", RFC 3031.
4. S. Black et al., "An Architecture for Differentiated Services," RFC2475, Dec. 1998.
5. K. Nichols, B. Carpenter "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification" RFC 3086.
6. C. Hopps "Analysis of an Equal-Cost Multi-Path Algorithm" RFC 2992, November 2000
7. S. Nelakuditi, Z.-L. Zhang, R.P. Tsang, and D.H.C. Du, Adaptive Proportional Routing: A Localized QoS Routing Approach, IEEE/ACM Transactions on Networking, December 2002.
8. N. Christin, J. Liebeherr and T. Abdelzaher. A Quantitative Assured Forwarding Service. In Proceedings of IEEE INFOCOM 2002.
9. S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Providing Absolute Differentiated Services for Real-Time Applications in Static-Priority Scheduling Networks," in IEEE/ACM Transactions on Networking, Vol. 12, No. 2, pp. 326-339, April 2004.
10. P. Aukia, M.Kodialam,P.Koppol, "RATES: A server for MPLS Traffic Engineering", IEEE network magazine, March 2000.
11. H. Mantar, I. Okumus, J. Hwang, S. Chapin " An Intra-domain Resource Management Model for Diffserv Networks", Journal of High Speed Networks, Vol. 15, pp.185-2005, 2006.

An Admission Control and TXOP Duration of VBR Traffics in IEEE 802.11e HCCA with Guaranteed Delay and Loss*

Tae Ok Kim[1], Yong Chang[2], Young-Tak Kim[3], and Bong Dae Choi[1]

[1] Department of Mathematics and Telecommunication Mathematics Research Center, Korea University, Korea
[2] Telecommunication Systems Division, Information & Communication Business, Samsung Electronics Co. Ltd, Korea
[3] Dept. of Information and Communication Engineering, Graduate School, Yeungnam University, Korea
violetgl@korea.ac.kr, yongchang@samsung.com, ytkim@yu.ac.kr, queue@korea.ac.kr

Abstract. We propose a new method to determine the optimal TXOP duration (TD) of scheduling algorithm for VBR traffics that guarantees delay and loss probability in IEEE 802.11e HCCA. In this algorithm, the TXOP Duration for each Traffic Stream (TS) is determined by the number (N) of packets to be transmitted during a TXOP duration, which is found by analyzing $M/D^N/1/N \times l$ queueing system where the delay bound of the TS is l times the Service Interval having D length. Numerical results show that the number of admitted TSs with guaranteed loss probability by new method is greater than the number of admitted TSs by other known schemes.

1 Introduction

IEEE 802.11 wireless LANs(WLANs)[1] is one of the most popular wireless technologies because of low cost and easy deployment. The explosive growth of real time and multimedia applications arose the problem of the required quality of service(QoS) of these applications such as guaranteed packet delay and packet loss probability. However 802.11 DCF protocol does not provide any QoS support.

Hybrid Coordination Function(HCF) in IEEE 802.11e[2] is to enhance the QoS Support by combining and extending the DCF and PCF of the MAC sublayer. The HCF consists of two channel access mechanisms : a contention-based channel access(EDCA) providing a probabilistic QoS support and a controlled channel access(HCCA) providing a parametric QoS support.

This paper is focused on a method to determine an optimal TXOP duration(TD) for the HCCA mechanism which provides a guaranteed loss probability

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

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 162–169, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and delay. As we will describe in more details in the next section, the 802.11e standard[2] proposes a reference design of the simple scheduler and admission control that is efficient for traffic with Constant Bit Rate(CBR) characteristic. However, a lot of applications such as video have VBR characteristic and the reference design would produce large loss probability for VBR traffic[4].

Several improvements for taking account of guaranteed loss probability were proposed. Ansel et al.[3] used a new scheduling algorithm called FHCF that utilizes difference between estimated average queue length at the beginning of the SI and ideal queue length. Fan et al. [4] proposed a scheme that decides TXOP duration by inversion of packet loss probability formula. Even if this scheme guarantees QoS, it has a tendency to overestimate TXOP duration.

In this paper, We propose a new method to determine the effective TXOP duration (TD) of scheduling algorithm for VBR traffics that guarantees delay and loss probability in IEEE 802.11e HCCA. In this algorithm, the TXOP Duration for each Traffic Stream (TS) is determined by the number (N) of packets to be transmitted during a TXOP duration, which is found by analyzing $M/D^N/N \times l$ queueing system where the delay bound of the TS is l times the Service Interval(SI) whose length is D. Numerical results show that the number of admitted TSs with guaranteed loss probability by our new method is greater than the number of admitted TSs by other known schemes.

The rest of the paper is organized as follows. Section 2 presents a reference design of the simple scheduler and admission control for completeness. Section 3 presents our new algorithm for calculation of an effective TXOP duration, Section 4 gives some numerical results to compare the other schedulers.

2 A Reference Design of the Simple Scheduler of HCF in the IEEE 802.11e

The IEEE 802.11e standard includes the reference design as an example scheduler. This scheduler uses the mandatory set of TSPEC parameters to generate a schedule: Mean Data Rate, Nominal MSDU Size and Maximum Service Interval (or Delay Bound). The Definitions of these parameters are as follows:

1. Mean data rate(ρ) : average bit rate for transfer of the packet, in units of bits per second.
2. Delay bound(D_b) : maximum delay allowed to transport a packet across the wireless interface (including queueing delay), in milliseconds.
3. Nominal MSDU size(L) : nominal size of the packets, in octets.
4. Maximum MSDU size(M) : maximum size of the packets, in octets.
5. PHY rate(R) : physical bit rate assumed by the scheduler for transmit time and admission control calculations, in units of bits per second.

2.1 An Example Scheduler

The first step of the reference design is the calculation of the Scheduled Service Interval(SI) of QAP. The scheduler chooses a number lower than the minimum of

all Maximum Service Intervals for all admitted streams, which is a submultiple of the beacon interval. This value will be the SI for all stations with admitted streams.

In the second step, the TXOP duration for a given SI is calculated for the stream. For the calculation of the TXOP duration for an admitted steam, the scheduler uses the following parameters: Mean Data Rate(ρ), Nominal MSDU Size(L) from the negotiated TSPEC, the Scheduled Service Interval(SI) calculated above, Physical Transmission Rate(R), Maximum allowable Size of MSDU, i.e., 2304bytes(M) and Overheads in time units(O). The Physical Transmission Rate is the Minimum PHY Rate negotiated in the TSPEC. If Minimum PHY Rate is not committed in ADDTS request, the scheduler can use an obseved PHY rate as R. In order to indicate the parameters of a specific TS "m", parameters are put subindex "m" like ρ_m and L_m. The TXOP duration is calculated as follows. First, the scheduler calculates the number N_m of MSDUs arriving during the SI for the traffic TS m with the Mean Data Rate ρ_m :

$$N_m = \left\lceil \frac{\rho_m * SI}{L_m} \right\rceil . \tag{1}$$

Then the scheduler calculates the TXOP duration(TD_m) of TS "m" as the maximum of (1)time to transmit N_m frames at rate R_m and (2) time to transmit one maximum size MSDU at rate R_m (plus overheads):

$$TD_m = \max(\frac{N_m \cdot L_m}{R_m} + O \ , \ \frac{M}{R_m} + O \ ) \ . \tag{2}$$

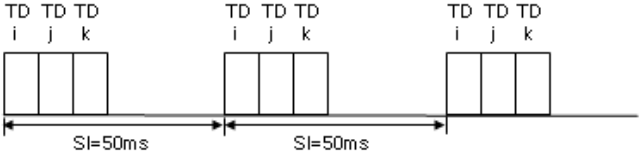


Fig. 1. Schedule for streams in the reference design

An example is shown in Figure 1. Three TSs with maximal service interval 50ms, 60ms, 100ms are admitted and the beacon interval is 100ms. Then, a Scheduled Service Interval(SI) is equal to 50ms using the steps explained above. Each TXOP duration of the TSs is used to serve only at most N_m packets among the packets arrived during the time interval from the moment of the beginning of the previous TXOP to the moment of beginning of current TXOP, which is equal to SI. Note that the packet arrivals greater than N_m are lost(see Fig.2).

2.2 An Admission Control Unit

This subsection describes a reference design for an admission control unit that administers admission of TS. When a new stream requests admission, the admission control process is done in three steps. First, the admission control unit

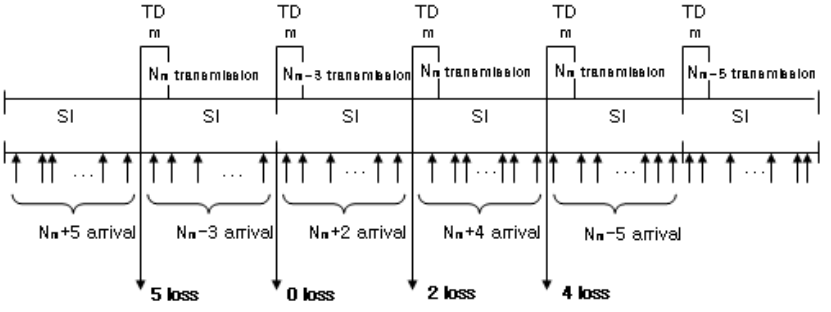


Fig. 2. Buffer state and behavior of packet loss in the reference design

calculates the number of MSDUs that arrive at the Mean Data Rate during the Scheduled Service Interval. Second, the admission control unit calculates the TXOP duration(TD) that needs to be allocated for the stream. Finally, the admission control unit determines that the stream can be admitted when the following in equality is satisfied:

$$\frac{TD_{k+1}}{SI} + \sum_{m=1}^{k} \frac{TD_m}{SI} \leq \frac{T_b - T_{cp}}{T_b} \tag{3}$$

where k is the number of existing streams and $k+1$ is used as index for the newly arriving stream. T_b indicates the superframe duration and T_{cp} is the time used for EDCA during the superfrme.

3 The Proposed Method for Determining an Effective TXOP Duration of VBR Traffic Which Guarantees Delay and Loss

In this section, we give a method to determine an effective TD of any TS so as to satisfy the required QoS conditions on delay and loss probability. The idea of our scheme is that we allow packets to wait until its own Delay Bound(D), and so we expect that TD to be assigned will be shorter than one given by Fan et al.[4] and the number of admitted TS is larger than one given by Fan et al.[4] These facts will be verified in our numerical examples. Different TSs have different delay bounds. For example, delay bound for TS_1 is 50ms and TS_2 is 100ms. Then, a Scheduled Service Interval(SI) is determined as 50ms by the same method with reference design. Packets of TS_2 can wait two consecutive SI instead of dropping packets not served on a SI. We assume that the number of packet arrivals on consecutive SIs are independent and identically distributed for this study. Packet size is assumed to be constant, for simplicity.

Let $l = \lfloor \frac{D}{SI} \rfloor$ be the number of TXOPs, which a packet of TS with delay bound D can be waited and transmitted after its arrival. SI is determined by the way

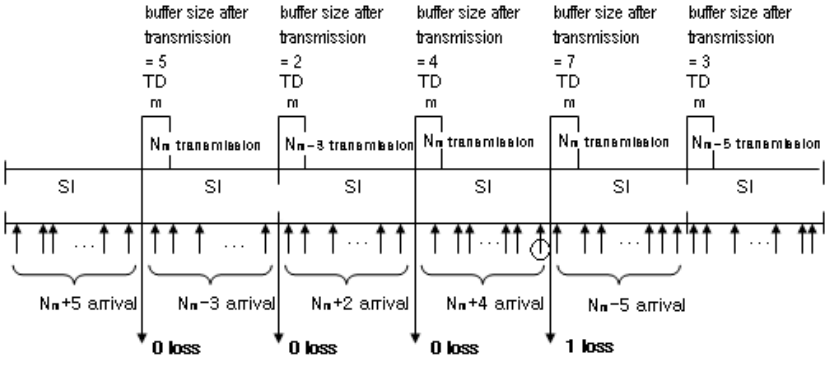


Fig. 3. Buffer state and behavior of packet loss in our scheme when $l = 2$ (N_m=7)

as explained in subsection 2.1. For instance, when $l = 2$, an arriving packet can be transmitted during only next two TXOPs, so that the packet will satisfy its delay requirement. Note that, in the reference scheme, an arriving packet either is sent during the next TXOP or is lost(Fig. 2). This characteristic intuitively implies that packet loss probability of our proposed scheme is less than that of the reference scheme. Fig. 3(N_m=7) shows the buffer state and behavior of packet loss when $l = 2$ in our scheme. As described in Fig. 3, the packets arriving at the time when the number of packets in the buffer of the QSTA is less than or equal to $N_m \times l$ will be transmitted, otherwise packets will be lost. So, we can describe the behavior of packet loss in our scheme by $GI/D^{N_m}/1/N_m \times l$ queueing system. Note that, by comparing Fig.2 and Fig 3, reference design has 11 packet's loss and our scheme has 1 packet's loss and so intuitively our scheme has less loss probability than reference design.

The packet loss probability of this queueing system is obtained as follows. Let X_k be the number of packets in the buffer at beginning of the kth SI and $a_i = \text{Prob}\{$ number of arrival packets in a SI $= \text{i}\}$. Then X_k is a discrete-time Markov chain whose one-step transition probability $P_{ij} = \text{Prob}\{\ \text{X}_{\text{k+1}} = \text{j} \mid \text{X}_\text{k} = \text{i}\ \}$ is given by

$$P_{ij} = \begin{cases} a_j & \text{if } \ i\in[0,N_m]\ ,\ j\in[0,N_m\times l-1] \\ 1-\sum_{r=0}^{N_m\times l-1} a_r & \text{if } \ i\in[0,N_m]\ ,\ j=N_m\times l \\ 0 & \text{if } \ i\in[N_m+1,N_m\times l]\ ,\ j\in[0,i-(N_m+1)] \\ a_{N_m+j-i} & \text{if } \ i\in[N_m+1,N_m\times l]\ ,\ j\in[i-N_m,N_m\times l-1] \\ 1-\sum_{r=i-N_m}^{N_m\times l-1} a_{N_m+r-i} & \text{if } \ i\in[N_m+1,N_m\times l]\ ,\ j=N_m\times l \end{cases} \tag{4}$$

Let $\pi_j = \lim_{k\to\infty} P\{X_k = j\}\ ,\ 0 \le j \le N_m \times l$, be the stationary distribution of the markov chain. Then the packet loss probability P_L is

$$\begin{aligned} P_L &= 1 - \frac{E(\text{number of transmitted packet in a SI})}{E(\text{number of arrived packet in a SI})} \\ &= 1 - \frac{j\cdot\sum_{j=0}^{N_m-1}\pi_j \ +\ N_m\cdot\sum_{j=N_m}^{N_m\times l}\pi_j}{E(\text{number of arrived packet in a SI})}. \end{aligned} \tag{5}$$

Then we can choose N_m as the minimum number that the packet loss probability is bounded by the required loss probability.

So, the TXOP duration for the TS is decided by

$$TD_m = \max(\ \frac{N_m \cdot L_m}{R_m} + O\ ,\ \frac{M}{R_m} + O\ )\quad . \tag{6}$$

After decision of TXOP duration by this procedure, the admission control process is done by the same way as the reference design.

4 Numerical Results

We obtain the numerical results in the same environments as [4] in order to compare our results with results of the reference design and the scheme in [4]. PHY and MAC parameters are summarized in table 1 for the numerical results. According to [2], the PLCP(Physical Layer Convergence Protocol) preamble and header are transmitted at Minimum Physical Rate to ensure that all stations can listen to these transmissions regardless of their individual data rates. We assume that the minimum physical rate in the analysis is 2Mbps and the t_{PLCP} is 96μs.

Table 1. PHY and MAC parameters

SIFS	10 us
PHY rate (R)	11 Mbps
Minimum PHY rate (R_{min})	2 Mbps

From [5], the bit rate of the ordinary streaming video over Internet is from 300kbps to 1Mbps. In the numerical analysis, we consider three mean data rate of video flow: 300kbps, 600kbps and 1Mbps. For each video stream of different mean

Table 2. Numerical Results for reference design and Fan's scheme[4]

		Reference scheme			Fan's scheme[4]		
ρ	L(byte)	N	TD(ms)	P_l	N	TD(ms)	P_l
300 Kbps	750	5	3.976	0.17547	10	7.953	0.00444
	1000	4	3.908	0.17228	8	7.817	0.00602
	1250	3	3.477	0.2240	7	8.112	0.00573
600 Kbps	750	10	7.953	0.12511	16	12.724	0.00547
	1000	8	7.817	0.11479	13	12.702	0.00529
	1250	6	6.953	0.160623	11	12.748	0.00579
1 Mbps	750	17	13.520	0.08789	23	18.291	0.00806
	1000	13	12.702	0.09397	18	17.588	0.00950
	1250	10	11.589	0.12511	16	18.543	0.00547

Table 3. Numerical Results for our scheme

		Our scheme (l=1)			Our scheme (l=2)			Our scheme l=3		
ρ	L(byte)	N	TD(ms)	P_l	N	TD(ms)	P_l	N	TD(ms)	P_l
300 Kbps	750	5	7.953	0.0044	6	4.771	0.0089	6	4.771	0.0010
	1000	8	7.817	0.0084	5	4.885	0.0095	5	4.885	0.0011
	1250	7	8.112	0.0057	5	5.794	0.0004	4	4.635	0.0011
600 Kbps	750	16	12.724	0.0055	11	8.747	0.0065	11	8.747	0.0007
	1000	13	12.702	0.0083	9	8.793	0.0073	9	8.793	0.0009
	1250	11	12.748	0.0058	7	8.112	0.0083	7	8.112	0.0010
1 Mbps	750	23	18.291	0.0096	18	14.314	0.0047	18	14.314	0.0006
	1000	18	18.564	0.0075	14	13.679	0.0056	14	13.679	0.0007
	1250	16	18.543	0.0055	11	12.748	0.0065	11	12.748	0.0007

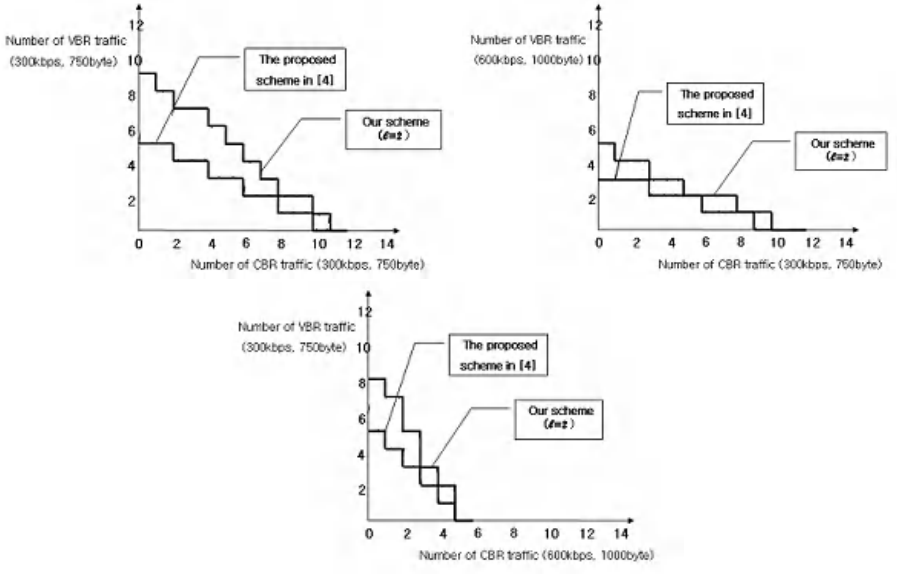


Fig. 4. Number of admitted TSs

data rate, different packet size are used in the analysis. Nominal MSDU size of each video stream are 750bytes, 1000bytes and 1250 bytes. The number of packet arrived during SI of VBR video source is assumed to be Poisson distributed. The video source is assumed to have a fixed packet length equal to Nominal MSDU size. The SI is 100ms and the proportion of the contention-free period in a SI is set to be half of the SI(i.e $\frac{T_b - T_{cp}}{T_b} = 0.5$). We assume that the packet loss probability requirement is 1%

The numerical results for other schemes and our scheme is presented in table 2 and 3 (N = the number of packets to be transmitted during a TXOP duration),

respectively. Table 2 quotes from [4]. It shows that the packet loss probability of the reference scheme for the VBR traffic is quite high, and consequently the reference scheme is not good for the VBR traffic. For proposed scheme in [4], the packet loss probability is bounded by 1%. However the required TXOP duration of this scheme is twice as long as one of the reference design. For our scheme, the packet loss probability is bounded by 1% and the TXOP duration is remarkably shorter than that of proposed scheme in [4]. Note that Fan's scheme[4] is essentially the same as our scheme with $l = 1$.

Fig. 4 represents the the admission regions of the proposed scheme in [4] and our scheme when there are two types of traffics, CBR traffics of 50ms delay bound and VBR traffics of 100ms delay bound. By this figure, we can recognize the fact that the admission region for our scheme is larger than one for the scheme in [4] and is a little bit less than the reference design. Therefore our proposed scheme is better than other two schemes in terms of packet loss probability and admission region.

References

1. IEEE, "International Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11:Wireless Medium Access Control(MAC) and Physical Layer(PHY) specifications," IEEE-802.11-1999, 1999
2. IEEE, "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11:Wireless Medium Access Control(MAC) and Physical Layer(PHY) specifications: Amendment 7: Medium Access Control(MAC) Quality of Service(QoS) Enhancements," IEEE P802.11e/D11.0,Oct. 2004
3. P. Ansel, Q. Ni and T. Turletti, "FHCF: A Fair Scheduling Scheme for 802.11e WLAN," INRIA Research Report No 4883, Jul. 2003
4. W.F. Fan, D.Y. Gao, D. H.K. Tsang and B. Bensaou, "Admission Control for Variable Bit Rate traffic in IEEE 802.11e WLANs," to be appeared in The Joint Conference of 10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications, Aug. 2004
5. F. Kozamemik, "Media Streaming over the Internet-an overview of delivery technologies," EBU Technical Review, Oct. 2002

NETSAQ: Network State Adaptive QoS Provisioning for MANETs

Shafique Ahmad Chaudhry, Faysal Adeem Siddiqui, Ali Hammad Akbar, and Ki-Hyung Kim*

Division of Information and Computer Engineering, Ajou University, Korea
{shafique, faysal, hammad, kkim86}@ajou.ac.kr

Abstract. The provision of ubiquitous services in Mobile Ad-hoc Networks (MANETs) is a great challenge, considering the bandwidth, mobility, and computational-resources constraints exhibited by these networks. Incorporation of modern delay-sensitive applications has made the task even harder. The traditional Quality of Service (QoS) provisioning techniques are not applicable on MANETs because such networks are highly dynamic in nature. The available QoS provisioning algorithms are either not efficient or are embedded into routing protocols adding a high computation and communication load. In this paper, we propose a Network State Adaptive QoS provision algorithm (NETSAQ) that works with many underlying routing protocol. It ensures the QoS provisioning according to the high level policy. NETSAQ is simple to implement yet minimizes the degradation of the best effort traffic at a considerable level. Our simulation results show that NETSAQ adapts well in MANET environments where multiple services are contending for limited resources.

Keywords: Mobile ad-hoc networks (MANETs), QoS, Network state Adaptive, QoS routing, Policy-based QoS provisioning.

1 Introduction

The role of Mobile Ad-hoc networks (MANETs) [1], in realization of a ubiquitous world, is manifold. MANET devices can serve as user interface, provide a distributed data storage, or act as a mobile infrastructure access point for other nodes. The MANETs are infrastructure-less networks that are envisioned to be spontaneously created whenever two or more nodes come in close proximity to each other. MANETs are characterized by dynamic topologies, limited processing and storage capacity, and bandwidth constrained wavering capacity links.

The described inherent characteristics of MANETs implicate newer requirements and technical challenges for the management of such networks. The incorporation of modern real time services, like transfer of audio and video concomitant to delay-agnostic services has further increased the management complexity. On one hand, MANET nodes have limited resources to share amongst contending services, on the other hand such services need and expect high priority on the network in order to meet higher user expectations about their reliability and quality.

* Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 170 – 179, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In order to meet QoS requirements, network managers can attempt to negotiate, reserve and hard-set capacity for certain types of services (hard QoS), or just prioritize data without reserving any "capacity setting" (soft QoS). Hard QoS can not be provided in ad hoc networks due to their dynamic nature. Therefore soft QoS [2] is provided in ad hoc networks using the QoS routing protocols and the IP Differentiated Services (DiffServ) framework. Mechanisms such as Integrated Services (IntServ) that emphasize on flow reservation cannot be implemented per se, in ad hoc networks because of the resource limitations and dynamic network conditions. DiffServ that provides aggregated classes of services may be a possible solution but necessitates some adaptation in order to be applicable in a completely dynamic topology.

Many QoS routing protocols have been proposed for wireless networks. One of the commonalities of these studies is that these routing protocols reserve the resources dynamically from source to destination before the application flow starts. If route breaks due to mobility, same procedure is repeated before a new route is established. Traditional MANET routing protocols do repair the broken path automatically but QoS violations can still occur even if the path is not broken due to mobility and interference etc. Very few QoS routing protocols for MANETs have considered this phenomenon such as [3]-[7]. The QoS routing protocols for MANETs are actually modifications of existing routing protocols to support QoS functionality. As a result, these QoS routing protocols utilize a lot of MANETs' resources and are heavier than traditional protocols [8].

In this paper we propose Network State Adaptive QoS provisioning algorithm (NETSAQ) that provides soft QoS without making resource reservations. It implements the high-level policies for QoS provisioning. NETSAQ is independent of routing protocols and can integrate with most of the routing protocols. On one hand it provides the high level control in form of policies and on the other hand it is independent of underlying routing protocol. Our simulation results show that its performance is a compromise between QoS routing and normal (best-effort) routing. NETSAQ avoids the degradation of best effort traffic, which is observed in many QoS routing schemes, while still providing minimum QoS guarantees for applications.

2 Related Work

Ensuring QoS through routing is relatively a new issue when it comes to MANETs. QoS routing protocols aim to search for routes with sufficient resources to satisfy initiating applications. The QoS routing protocols work closely with the resource management module in order to set up paths through a list of mobile nodes to meet the end-to-end service requirements in terms of bandwidth, delay, jitter, and loss, etc. The computational and communication cost of QoS routing is known to be fairly high and it has raised the questions whether or not should it be tackled in MANETs.

Many proposals have been made for QoS provisioning in MANETs. The In band Signaling (INSIGNIA) [3], Stateless Wireless Ad hoc Networks (SWAN) [4], Core Extraction Distributed Ad Hoc Routing (CEDAR) [5], Adaptive Source Routing (QoS-ASR) [6], and Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR) [7] are examples of QoS-routing proposals. We have summarized

various parameters to have an insight into the important features provided by these protocols. Table 1 describes the distinguishing features of these protocols.

Table 1. Comparison with QoS routing protocols

	QOLSR	QoS-ASR	INSIGNIA	SWAN	NETSAQ
QoS type	Fixed	Fixed	Dynamic (degrade to best effort)	Fixed	Dynamic (degrade to min QoS)
QoS parameters	Delay, bandwidth	Delay, bandwidth, packet loss	Bandwidth	Bandwidth	Delay, bandwidth, packet loss
Admission Control	Yes	Yes	No	Yes	Yes
Resource Reservation	Priori soft and posteriori hard reservation	Priori hard reservation	Posteriori soft reservation	Priori hard reservation	Posteriori hard reservation
Complexity	High	Highest	Low	High	Low
Application level support	No	No	Yes	No	Yes
Nodes doing rerouting due to interference	Source and intermediate nodes	Source	Source and intermediate nodes	Source	Source
Nodes measuring QoS	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source, destination and intermediate nodes	Source and destination
QoS violation rerouting	Path break and path degradation	Path break and path degradation	Path break	Path break and path degradation	Path break and path degradation
Best effort traffic drop probability	High	Highest	Lowest (admitted)	Low (rate decreased gradually)	Low (Adaptive QoS)

3 System Model and Assumptions

The system model for NETSAQ can best be implemented by an autonomous policy based network management architecture that provides a multi-tier management [9]. In this work we have proposed an architecture for cluster based MANETs. Each cluster head is equipped with a Domain Policy Manager (DPM). The DPM is responsible to monitor the overall status of the MANET; making, updating and distributing the network wide policies for the cluster and correspondingly deploy policies on the MANET nodes. The DPM's architecture is shown in fig.2 (a). For simple MANET nodes, which may exhibit great degree of heterogeneity, are assigned an extended local agent (ELA) or a simple local agent (SLA). These local agents are responsible to implement the policies on the nodes they are deployed at. The components of an ELA are shown in fig. 2(b).

Many QoS policies are defined for various services. Each policy can be associated with specific types of applications or users or even to a specific user using specific application.

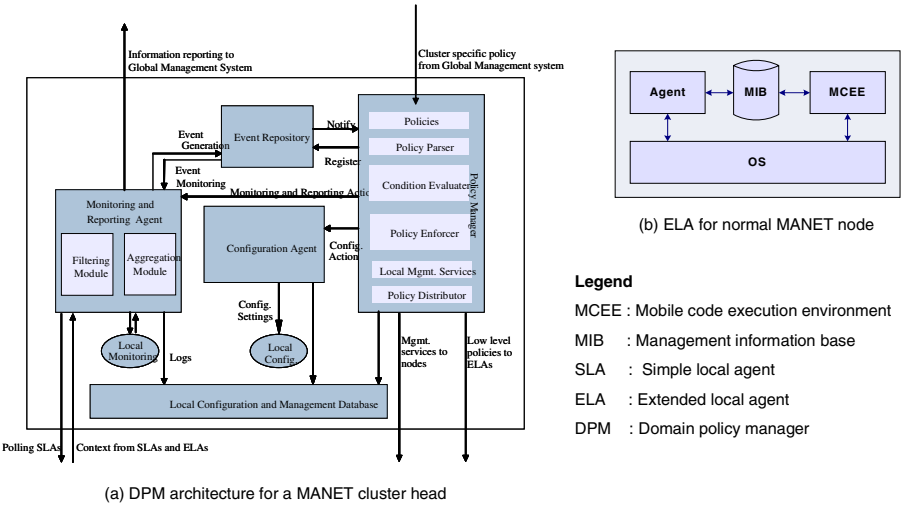


Fig. 1. Management components for cluster head and normal MANET node

In order to provide QoS to specific application, NETSAQ algorithm is executed and dynamic QoS is provided to the specific traffic according to the policy. The application specifies a minimum acceptable and maximum allowed QoS values called b_{min} and b_{max} respectively. During adverse network conditions, at least b_{min} is satisfied but if the network state is good then a higher level of QoS is provided. Thus QoS depends upon the network state and is automatically adjusted. It is assumed that the nodes exchange information like packet loss ratio, delay and available bandwidth when building and maintaining their routing tables. This can be done through *hello messages* which most of the protocols exchange for making and updating their routing tables.

3.1 Node Mobility and Interference Scenario

Consider a network that provides services to users that belong to different groups such as special and ordinary groups. Now a MANET special user starts a video conferencing application. The QoS of the application can be violated due to: a) *mobility,* whenever the source, the destination or any intermediate node moves, the established QoS path can be broken, b) *interference or contention,* whenever the number of nodes around the QoS path increase considerably, interference increases and contention against wireless channels escalates. In both the situations there is a possibility of QoS degradation even if the resources were reserved beforehand. In either case, the alternative path is needed which can facilitate the required QoS provisioning. This phenomenon is inherent to MANETs, and cannot be eliminated.

4 Network State Adaptive QoS Provisioning Algorithm

In a network, that renders services to multiple user applications, some applications may spell their QoS. We assume that, any application that starts has certain QoS

requirements with minimum and maximum QoS bandwidth constraints b_{min} kbps and b_{max} kbps. In order to admit this application into the network, the average data rate of the network is checked. A bandwidth threshold s, based on network media access delay D and Packet Loss Ratio (PLR) is calculated for the application at the initiating node of the application. The resources are then marked along the path based on s, i, and if such bandwidth is not available then we use b_{min}. Once all the links along the path, from source to destination are covered, then we start the application flow. After the flow is established, resources can be reserved on intermediate nodes according to threshold s. This, posteriori reservation, is different from reserving the resources before starting the flow (priori reservation). Posteriori reservation eliminates the complexity and delay which is inherent to priori reservation and provides better response time to the application user. The application flow is re-routed if the end-to-end bandwidth falls below the reserved resources for more than a specific time. The algorithm is depicted in fig. 4.

Lets i kbps be the initial averaged data rate over t sec

Set a threshold s kbps using network media access delay D and packet loss ratio PLR

$s = max [\{1- 0.5 (w_1 D + w_2 PLR)\} b_{max} , b_{min}]$

(Where w_1 and $w2$ are the weights associated to D and PLR)

Reserve the bandwidth along the path on every node equal to s kbps

If s kbps bandwidth can't be reserved, reserve i kbps

If even i kbps bandwidth is not available, reserve b_{min} kbps

Reroute if the end-to-end QoS falls below the reserved resources for more than t sec

Calculate value of s before rerouting

New route is found with resources greater than or equal to s kbps

If no such path exists, find a path with resources greater than or equal to i kbps (only if $s > i$)

In the worse case, find a path with resources greater than or equal to b_{min} kbps

Fig. 2. NETSAQ algorithm

If the established path is broken due to link failure or mobility of any node(s), rerouting is done at the node where the path is broken. If this intermediate node can't find the QoS path, the next upstream node tries to find the path. If no QoS path can be established from the upstream node as well, source is notified to broadcast a new route request with b_{min} kbps and b_{max} kbps QoS constraints.

If the data rate for the application decreases than the reserved bandwidth for more than t sec, the destination will notify the source. It means all the intermediate nodes do not need to continuously monitor the data rate violations. Also, in case of interference, the source has a greater possibility to find new disjoint routes. The routing agent at the source will then be invoked to find a new route with data rate > s kbps or > i kbps or > b_{min} kbps as it would be described in the policy.

MANET routing protocols may use link level acknowledgement (ACK) messages and a timeout period for link level connectivity information. In an area with dense

population of nodes, hidden terminal problem can become quite significant. Due to hidden terminal problem and high contention, some nodes will not receive the link layer ACK packets from the neighboring nodes. When the timeout period expires, a node declares the link as broken, discards all ongoing communication packets and generates a route error message [10]. This causes the throughput to drop drastically. The communication resumes when a new path is found or the same path is eventually re-discovered. This instability problem is caused by fast declaration of link failures which is rooted at the link layer. The breaking and rediscovery of the path result in the drastic throughput oscillations. In order to avoid this problem, we extend the solution proposed in [11]. This solution uses a "don't-break before-you-can-make" strategy. This strategy is based on modifying the routing algorithm so that the routing agent continues to use the previous route for transmissions before a new route can be found. When the new route is found or the same route is eventually re-discovered, all nodes discard the previous route and switch to the new one (or the same one) for transmissions. An example for the explanation is shown is fig.3.

Let b_{min} = 50 kbps, b_{max} = 100 kbps, t = 1 min, w_1D = 0.2, w_2PLR = 0.3 and i=65 kbps

Then threshold s:

s = max [{1- 0.5 (w1 D + w2 PLR)} b_{max}, b_{min}]

s = max [{1- 0.5 (0.2 + 0.3)} 100, 50] = 75 kbps

Reserve the resources along the path equal to 75 kbps (s)

Else reserve the resources equal to 65 Kbps (i)

Otherwise reserve the resources equal to 50 Kbps (b_{min})

If resources are reserved equal to 75 kbps

- when destination node receives data rate < 75 kbps for 1 min
- It will notify the source
- Source will establish a new route with data rate > s kbps
- If no such route is available, a route with data rate > i or minimum bandwidth b_{min} condition will be set up

If resources are reserved equal to 65 kbps

- when destination node receives data rate < 65 kbps for 1 min
- It will notify the source
- Source will establish a new route with data rate > s kbps
- If no such route is available, a route with data rate > i or minimum bandwidth b_{min} condition will be set up

If resources are reserved equal to 50 kbps

- when destination node receives data rate < 50 kbps for 1 min
- It will notify the source
- Source will establish a new route with data rate > s kbps
- If no such route is available, a route with data rate > i or minimum bandwidth b_{min} condition will be set up

Fig. 3. A numerical example for NETSAQ algorithm working

5 Performance Evaluation

The simulations is this section evaluate the suitability of the algorithm to support adaptive flows in a MANET under various conditions such as traffic, mobility, and channel characteristics. In particular, we evaluated system wide adaptation dynamics and the impact of threshold based rerouting mechanisms and mobility on end-to-end sessions.

We used OPNET simulator and the simulation environment consists of 20 Ad-hoc nodes in an area of 500m x 500m. Each mobile node has a transmission range of 300m and shares a 5.5 Mbps air interface between neighboring mobile nodes. The nodes follow the standard random waypoint mobility model with the maximum speed of 10m/s used. The underlying routing protocol is Dynamic Source Routing (DSR). We have used various application flows having different bandwidth requirements ranging from 90 Kbps to 320 Kbps. An arbitrary number of best effort flows are randomly generated to introduce different loading conditions distributed randomly throughout the network. These loading flows are dynamic in nature which helps analyze the adaptive behavior of NETSAQ.

We measure per-session and aggregate network conditions for a number of experiments that analyze adaptation, threshold based rerouting, and nodes mobility. We observe throughput, delay, packet loss, rerouting frequency and degradation, as the measures of system dynamics during the course.

5.1 Adaptive Flows

We measure the performance of two adaptive flows with User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Fig. 4 and 5 show flows with b_{max} values as 160 and 320 while b_{min} values are 90 and 160 respectively.

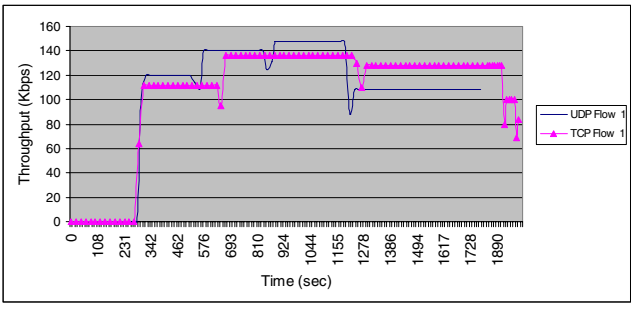


Fig. 4. Adaptive flow 1

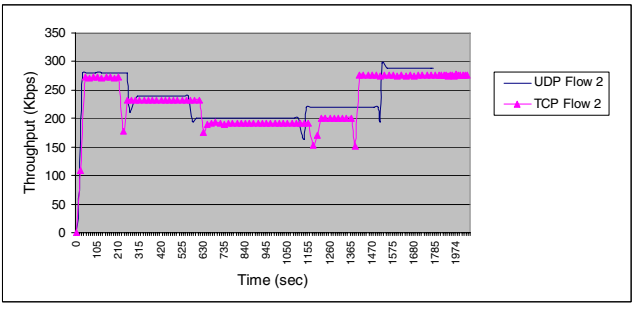


Fig. 5. Adaptive flow 2

Variable traffic load was generated at random nodes that caused the flows to be rerouted depending on the dynamic network conditions. Momentary QoS violations occurred due to rerouting but overall the flow kept its minimum bandwidth guarantee. We observed that the flow fluctuated between its maximum and minimum bandwidth constraints. Especially the flows under TCP fall below minimum bandwidth constraint at times. This is due to the TCP rate adjustment because of large network delay and packet loss.

5.2 Increasing Number of Nodes

The simulation environment consists of 5, 10 and 15 ad hoc nodes in an area of 200m x 200m. Each mobile node has a transmission range of 100m and shares 1Mbps air interface between neighboring mobile nodes within that transmission range. Two video streams were introduced in the network with different QoS requirements. Stream 1 has minimum QoS requirement as 150 Kbps and maximum QoS requirement as 300 Kbps. Similarly, stream 2 has minimum QoS requirement as 40 Kbps and maximum QoS requirement as 100 Kbps. As we compare Fig. 6 and Fig. 7 it is obvious that NETSAQ adapts to the increasing number of nodes very well. As we increase the number of nodes from without any QoS mechanism, the interference and contention increase considerably and degrade the performances of both the streams.

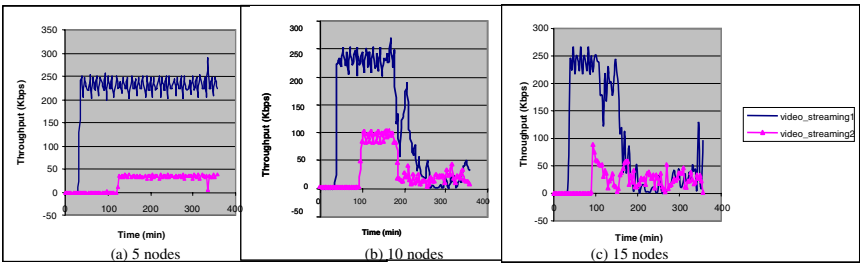


Fig. 6. Increasing number of nodes with no QoS

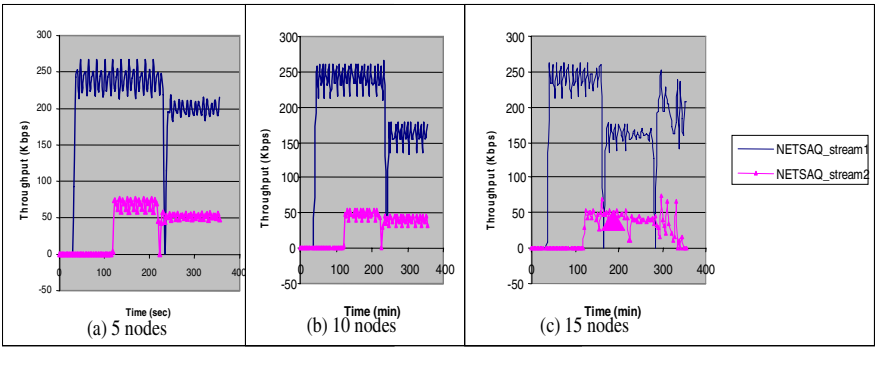


Fig. 7. Increasing number of nodes with NETSAQ

The Fig. 7 shows NETSAQ adapts to the changing conditions and finds a route with a slightly decreased QoS value within the minimum and maximum QoS constraints for both the streams. NETSAQ reroutes both the streams to a lower QoS levels to cope up with the degrading network conditions, but when the loading traffic increases and saturates the network towards the end, only best effort QoS can be provided.

5.3 Changing Transmission Rate

We evaluated the performance under increasing transmission rates (by changing the air interface and increasing stream rates as well. The simulation environment consists of 15 ad hoc nodes in an area of 200m x 200m. Each mobile node has a transmission range of 100m. The transmission rate or air interface is changed from 1 to 2 and 5.5 Mbps. The comparison between Fig. 8 and Fig. 9 allows us to observe the way NETSAQ adapts to transmission rate variations.

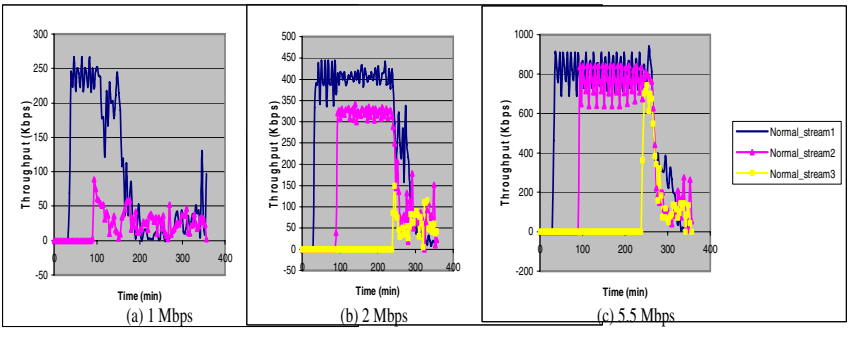


Fig. 8. Different transmission and streaming rates with no QoS

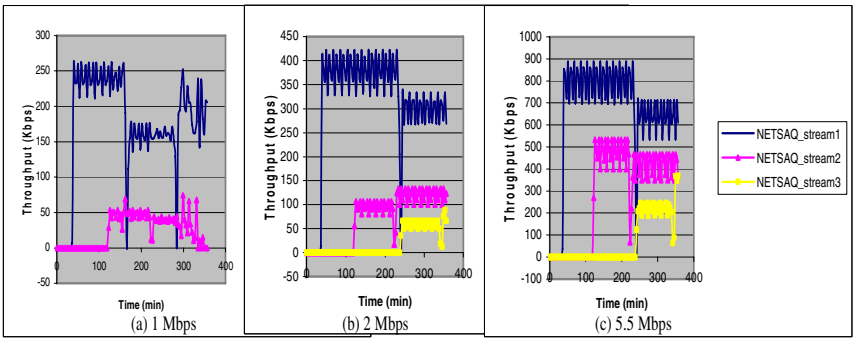


Fig. 9. Different transmission and streaming rates with NETSAQ

6 Conclusion

In this paper we have proposed NETSAQ, a network state adaptive QoS algorithm for MANETs. NETSAQ implements the QoS provisioning policies using any underlying

routing protocol. It eliminates the bulky computation and communication for QoS in routing algorithm. It utilizes lesser MANET resources as compared to existing QoS routing protocols. It is simple yet minimizes the degradation of best effort traffic, which is observed in many QoS routing schemes.

References

1. Frodigh, M., Johansson, P., Larsson, P.: Wireless ad hoc networking: the art of networking without a network. *Ericsson Review*, No.4, pp. 248-263, 2000.
2. Veres, A., Campbell, A., Barry, A., Sun, L. H.: Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control. *IEEE JSAC*, vol. 19, no. 10, Oct. 2001.
3. Lee, S., Ahn, G., Zhang, X., Campbell, A.: INSIGNIA: *IETF Internet Draft*, draft-ietfmanet-insignia-01.txt, work in progress, Nov. 1999.
4. Ahn, G., Campbell, A., Veres, A., Sun, V.: SWAN, *IETF Internet Draft*, draft-ahn-swanmanet-00.txt, work in progress, Oct. 2002.
5. Sivakumar, R., Sinha, P., Bharghavan, V.: Core Extraction Distributed Ad Hoc Routing (CEDAR) Specification. *IETF Internet Draft*, draftietf-manet-cedar-spec-00.txt, work in progress, 1998.
6. Labiod H., Quidelleur, A.: QoS-ASR: An Adaptive Source Routing Protocol with QoS Support in Multihop Mobile Wireless Networks. *IEEE VTC'02*, pp. 1978- 1982. 2002.
7. Badis, H., Al-Agha, K.: Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR). *IETF Internet Draft*, work in progress draft-badis-manet qolsr-01.txt, Apr. 2005
8. Ge, Y., Kunz, T., Lamont, L.: Proactive QoS Routing in Ad Hoc Networks. *ADHOC-NOW'03*, pp. 60-71, 2003.
9. Chaudhry, S.A., Akbar, A.H., Siddiqui F.A., Sik Y.W.: Autonomic Network Management for u-Zone Networks. *UbiCNS' 05*, Jeju, Korea, June 9-10, 2005
10. Charles E. Perkins and Elizabeth M. Royer, "The Ad hoc On-Demand Distance Vector Protocol" In Charles E. Perkins, editor, *Ad hoc Networking*, pages 173–219. Addison-Wesley, 2000.
11. Chung-Ng, P. Liew, K.: Re-routing Instability in IEEE 802.11 Multi-hop Ad-hoc Networks. *WLN'04*, Nov. 2004.

End-to-End QoS Guaranteed Service in WLAN and 3GPP Interworking Network

Sung-Min Oh[1], Jae-Hyun Kim[1], You-Sun Hwang[2], Hye-Yeon Kwon[2], and Ae-Soon Park[2]

[1] Ajou University, Korea
smallb01@ajou.ac.kr, jkim@ajou.ac.kr
[2] Electronics and Telecommunications Research Institute, Korea
ys3838@etri.re.kr, hywon@etri.re.kr, aspark@etri.re.kr

Abstract. In this paper, we model the end-to-end QoS provisioning mechanisms in the WLAN and 3GPP interworking network. For the end-to-end QoS guaranteed service, we model the control plane and user plane considering the WLAN and 3GPP interworking network. And we propose the QoS parameter/class mapping and DPS packet scheduler. By the simulation results, DPS can provide the low end-to-end delay of voice traffic, even though a traffic load increases by 96%. Especially, the end-to-end delay of voice is much smaller when the QoS parameter/class mapping and DPS are applied to the interworking network.[1]

1 Introduction

In recent years, many mobile users are demanding anytime and anywhere access to high-speed multimedia services for next generation communication system. There are many number of communication technologies for next generation system. In order to satisfy the user requirements for the wireless local area network (WLAN) and third generation partnership project (3GPP) interworking network, the 3GPP is concerned about the WLAN and 3GPP interworking [1].

3GPP has been studying and standardizing the WLAN and 3GPP interworking mechanism. However, it is insufficient to investigate the quality of service (QoS) provisioning technology in the WLAN and 3GPP interworking network[2].

There are various challenges to provide the end-to-end QoS guaranteed services through WLAN and 3GPP interworking network [3], [4]. First, there are many differences between their QoS provisioning technologies such as the QoS parameters, service classes and so on. Accordingly, the mapping mechanism is required for a seamless service. Second, a bottleneck may be generated due to the limited capacity and the overload at a gateway linked with backbone network. In order to solve these problems, we define the functional features based on the end-to-end QoS architecture in WLAN and 3GPP interworking network. We also propose the new QoS provisioning technologies, and then analyze the performance of the proposed mechanism using a simulator.

[1] This work was supported by the second stage of Brain Korea 21 (BK21) Project in 2006.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 180–189, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 WLAN and 3GPP Interworking Network Proposed by 3GPP

3GPP working groups are enthusiastic about the standardization of the WLAN and 3GPP interworking technologies. They have proposed standard for the WLAN and 3GPP interworking network [2], [5], [6].

2.1 WLAN and 3GPP Interworking Network Architecture

3GPP TR 23.882 defines the WLAN and 3GPP interworking network architecture as shown in Fig. 1. In this network architecture, WLAN is interconnected with 3GPP network based on universal mobile telecommunication system (UMTS) network. The network elements are added to WLAN network to link up with 3GPP network such as WLAN access gateway (WAG) and packet data gateway (PDG). WAG allows visited public land mobile network (VPLMN) to generate charging information for users accessing via the WLAN access network (AN) in the roaming case. WAG filters out packets based on unencrypted information in the packets. PDG is to directly connect to 3GPP data service network. PDG has responsibilities that it contains routing information for WLAN-3GPP connected users and performs address translation and mapping. PDG also accepts or rejects the requested WLAN access point name (W-APN) according to the decision made by the 3GPP AAA Server. In the 3GPP standards, they define the additional WLAN networks as the WLAN Direct IP network and WLAN 3GPP IP access network. The WLAN Direct IP network is directly connected to internet/intranet, and the WLAN 3GPP IP access network including WAG and PDG is connected to 3GPP network [6].

2.2 Research Issues for End-to-End QoS Provisioning

In WLAN and 3GPP interworking network architecture, there are various research issues for the end-to-end QoS guaranteed service.

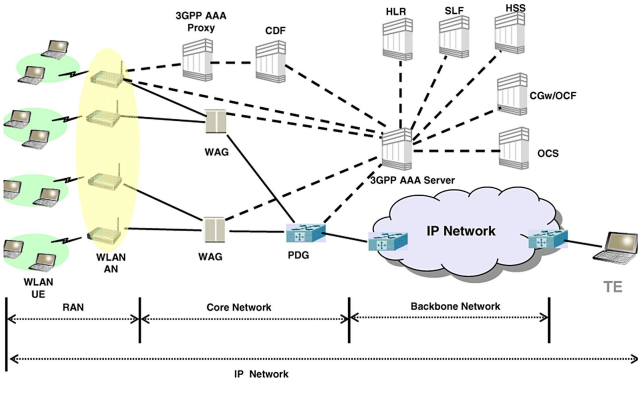


Fig. 1. WLAN and 3GPP interworking end-to-end network architecture

It is a very important issue to map the QoS provisioning mechanisms of WLAN and 3GPP. When a WLAN UE transmits packets to a 3GPP subscriber, the QoS provisioning mechanisms of WLAN and 3GPP may not be worked together due to the absence of QoS information of transmitted packets. Especially, the QoS parameter mapping function should be located in the PDG, since the PDG is the edge node linked with 3GPP network.

The other issue is that a bottleneck can be occurred in the PDG when WLAN UEs may forward many packets to a PDG. The bottleneck can be a serious problem, since the end-to-end delay of voice or video traffic can be rapidly increased by the bottleneck. Therefore, we propose two QoS provisioning mechanisms which are the QoS parameter/class mapping function and packet scheduler.

3 Proposed QoS Provisioning Mechanisms

In the conventional WLAN and 3GPP interworking standard, the QoS provisioning mechanisms have not been considered in detail [2]. Therefore, we define the functional features for the end-to-end QoS provisioning based on the end-to-end QoS architecture and model the control and user plane considering the detailed functional features. Besides the control and user plane, we propose QoS parameter/class mapping and packet scheduler (DPS).

3.1 End-to-End QoS Architecture and Functional Features

As shown in Fig. 2, the end-to-end QoS architecture was modeled by 3GPP [2]. However, the functional features are not defined in the 3GPP standard. We define the functional features of each element based on the end-to-end QoS architecture as follows. End-to-End Service provides the end-to-end QoS guaranteed service through the 3GPP IP Access Bearer Service and External Bearer Service. 3GPP IP Access Bearer Service includes the WLAN Bearer Service, since the WLAN Bearer Service can provide a QoS guaranteed service in WLAN. For the QoS guaranteed service in WLAN, we consider the IEEE 802.11e. The External Bearer Service provides the QoS guaranteed service in backbone network. In backbone network, we consider two QoS provisioning mechanisms, such as a

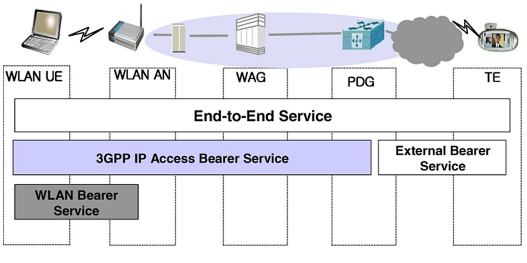


Fig. 2. End-to-end QoS architecture in WLAN and 3GPP interworking network

differentiated service (DiffServ) and integrated service (IntServ). When the DiffServ mechanism is applied to the backbone network, the PDG has to perform a DiffServ edge function.

3.2 Control and User Plane

Based on the end-to-end QoS architecture, we model control and user plane as shown in Fig. 3 and Fig. 4. In control plane, we define the functional blocks which manage the signal procedure for the resource reservation. In user plane, we define the functional blocks that are needed for QoS guaranteed service when packets are transmitted through the WLAN and 3GPP interworking network. The definition of the functional features is as follows.

Functional Features in Control Plane. IP bearer service (BS) Manager allocates the remote IP address for transmitting packets through end-to-end network. Translation Function is applied to WLAN UE and PDG. In WLAN UE, Translation Function translates the QoS information of application data to that of IEEE 802.11e in forward link and vice versa in reverse link. In PDG, Translation Function translates the QoS information of external network to that of 3GPP IP access BS network in forward link and the opposite in reverse link. The Admission/Capability Control is located in the WLAN UE, PDG and WLAN AN. In the WLAN UE and PDG, the Admission/Capability Control decides whether to admit the received call. In the WLAN AN, the call admission control (CAC) is applied to manage the wireless resource in IEEE 802.11e. 3GPP IP Access BS Manager requests the QoS information to Translation Function or Admission/Capability and manages the tunneling protocol. Access Network Manager is for routing according to the local IP address. WLAN BS Manager requests the QoS information considering the wireless resource in WLAN network, and interrogates the available resource. WLAN BS Manager manages the negotiation process of traffic specification (TSPEC) defined in IEEE 802.11e. WLAN PHY BS Manager manages the bearer service according to the wireless environment.

Functional Features in User Plane. In user plane, we define the functional features when packets are transmitted through WLAN 3GPP IP access network. Classification Function classifies the packet which is received from the external network or application layer. Traffic Conditioner controls the uplink or downlink traffic using the QoS information. Mapping Function performs the service class mapping of QoS parameters among WLAN, IP and 3GPP. Packet Scheduler rearranges the transmission order according to the service class.

3.3 QoS Provisioning Technologies

For the QoS provisioning, we propose the QoS parameter/class mapping technology and DPS packet scheduler based on the control plane and user plane. The QoS parameter and class mapping technologies are used to translate the

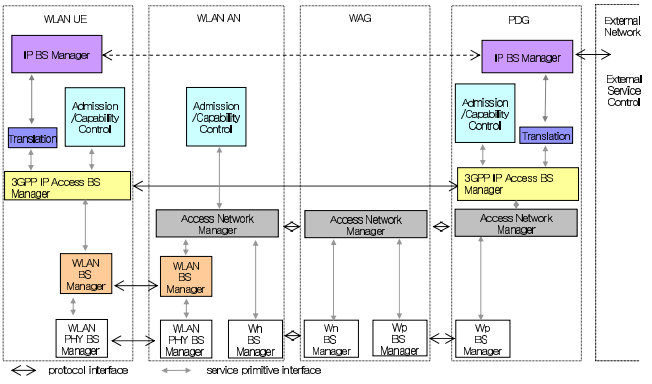


Fig. 3. Control plane based on the end-to-end QoS architecture

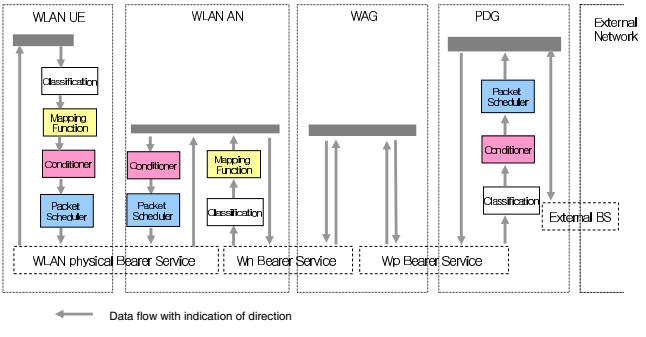


Fig. 4. User plane based on the end-to-end QoS architecture

QoS information in the control plane and user plane. The packet scheduler is applied to manage the data flow in user plane. We evaluate the performance of the QoS parameter/class mapping algorithm and packet scheduler in section 4.

QoS Parameter Mapping. The QoS parameter mapping function is located in WLAN UE and PDG. It translates between the QoS parameters of WLAN and 3GPP. There are many QoS parameters which are similarly defined by WLAN and 3GPP. For example, Maximum Bit-Rate and Maximum service data unit (SDU) Size defined in 3GPP is similar to the Peak Data Rate and Maximum MAC service data unit (MSDU) Size defined in WLAN respectively. Therefore, we propose the QoS parameter mapping table considering the relation of the QoS parameters of WLAN and 3GPP as shown in Table 1.

QoS Service Class Mapping. We should consider QoS service class mapping among WLAN, IEEE 802.1D (IP) and 3GPP for the packet scheduling, because the data packet would be transmitted through WLAN, IP, and 3GPP network.

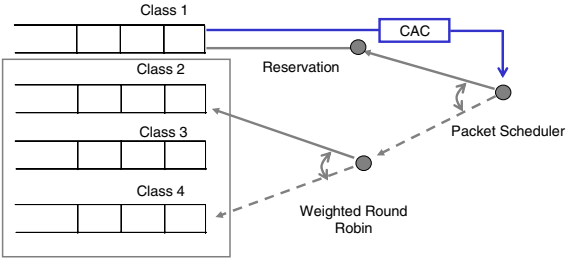


Fig. 5. DPS

Table 1. The QoS parameter mapping of WLAN and 3GPP

3GPP QoS parameters(3GPP TS 23.107)	WLAN QoS parameters(TSPEC)
Maximum bit rate (kbps)	Peak data rate (bps)
Maximum SDU size (octects)	Maximum MSDU size (octects)
SDU format information	Burst size (octects)
Transfer delay (ms)	Delay bound (μsec)
Traffic handling priority	User priority

Table 2. The service class mapping of WLAN, 3GPP, and IP

802.1D	3GPP	WLAN
7,6	Conversational	Continuous time QoS traffic (HCCA)
5,4	Streaming	Controlled-access CBR traffic (HCCA)
0,3	Interactive	Bursty traffic (HCCA)
2,1	Background	Unspecified non-QoS traffic (HCCA)

Since the QoS service classes of them are defined for each service requirement, we can make the QoS service class mapping table considering the similar service requirements like Table 2.

Dynamic Processor Sharing (DPS). The DPS is proposed for the QoS guaranteed service. DPS can keep the delay bound of voice traffic by the resource allocation for voice traffic. In addition, DPS can provide the fairness for the others. The DPS is shown in Fig 5 and the explanation of DPS operation is as follows. We define the four service classes as class1, class2, class3, and class4 according to the priority. Class1 indicates the highest priority traffic class which is very sensitive to delay such as voice of internet protocol (VoIP). To support the class1, the resource manager allocates the resource to guarantee the delay bound when the class1 is generated. Class2 is also sensitive to delay, but the tolerance of delay is larger than class1 like video streaming. Class3 and class4 are insensitive to delay, but they are critical to the packet drop probability such as hyper text transfer protocol (HTTP) and file transfer protocol (FTP). Therefore, the weighted round robin (WRR) is applied to the class2, class3,

Table 3. The service traffic models

Service class	Parameters
Voice	Voice encoder scheme : G.711
	PHY throughput of a voice user : 174 kbps
	Silence : exponential (0.65)
	Talk spurt : exponential (0.35)
	Session duration : constant (30 sec)
Video	Frame inter-arrival time : 10 frame/sec
	PHY throughput : 1Mbps, 1.3Mbps, 1.5Mbps, and 1.7Mbps
FTP	Inter-request time : exponential (30 sec)
	File size : 5000 bytes
HTTP	HTTP spectification : HTTP 1.1
	Page inter-arrival time : exponential (60 sec)
	Number of objects : constant (6)

class4 to keep the QoS and fairness. The weight should be selected considering the traffic load for the efficiency of capacity. Since class1 occupies the resource, the fairness problem may be occurred if the resource limitation is not defined. To solve this fairness problem, the CAC is applied to the class1.

4 Performance Evaluation

In this section, we present the network model and results of the performance evaluation for our proposed QoS provisioning technologies. We built the simulator for the QoS provisioning technologies using OPNET.

4.1 Network Model

For the performance evaluation, we simply model the end-to-end reference network architecture which consists of WLAN UEs, three APs, a WAG, a PDG, a gateway, and an internet server. The WLAN UE and AP include the IEEE 802.11, but we do not consider the IEEE 802.11e in this simulation. We connect the PDG with the gateway by a PPP-E1 link for implementing the bottleneck phenomenon. We can expect that the performance of the system rapidly decreases as the total traffic load approaches about 2Mbps.

To increase the traffic load in this simulation, we increase the number of voice users and video traffic load. We model the service traffic based on the traffic models provided by OPNET as shown in Table 3. Since we are interested in the performance of the QoS parameter/class mapping and DPS, we compare the various packet schedulers applied to the PDG. We consider the packet schedulers as first in first out (FIFO), WRR, strict priority (SP), and DPS. We set up the weight ratio according to the traffic load ratio for DPS and WRR.

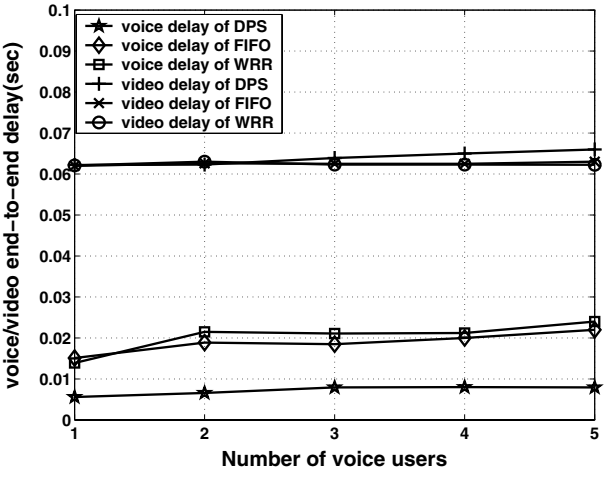


Fig. 6. End-to-end delay of voice and video traffic according to the packet schedulers and the number of voice users

4.2 Simulation Results

We present our simulation results to evaluate the DPS through comparing it with FIFO, WRR, and SP.

In our simulation, we evaluate packet schedulers for three cases. First, we analyze the voice/video end-to-end delay with increasing a voice traffic load. In Fig. 6, a total traffic load increases as 62.3%, 70.8%, 79.33%, 87.8%, and 96.3% according to the increase of the number of voice users. Fig. 6 presents that DPS can provide the low voice end-to-end delay under 10msec even though a traffic load increases above 96.3%. The reason is that DPS gives the highest priority for the voice packet. We can also analyze that DPS is not poor at the video end-to-end delay from Fig. 6, because the performance difference between others is very small. Therefore, we found that DPS is suitable for the packet scheduler of PDG irrespective of the voice traffic load.

Second, we analyze the voice/video end-to-end delay with increasing a video traffic load. In this case, we can confirm the robustness of DPS according to the increase of the video traffic. Since DPS gives a highest priority to a voice packet, the voice end-to-end delay does not increase though the video traffic increases. In Fig. 7, when a PHY throughput of video traffic increases from 1Mbps to 1.7Mbps, the voice end-to-end delay of DPS does not change at all. And there are very few differences among packet schedulers for the video end-to-end delay. Therefore, DPS is extremely suitable to be applied to the packet scheduler of the PDG compared with FIFO and WRR. The last case is the comparison between SP and DPS. SP is very similar to DPS, because SP also gives a highest priority to a voice packet when a voice packet arrives at the PDG. However, SP can not guarantee the fairness when the number of voice users rapidly increases. If a number of voice users transmit a number of voice packets to the PDG, the

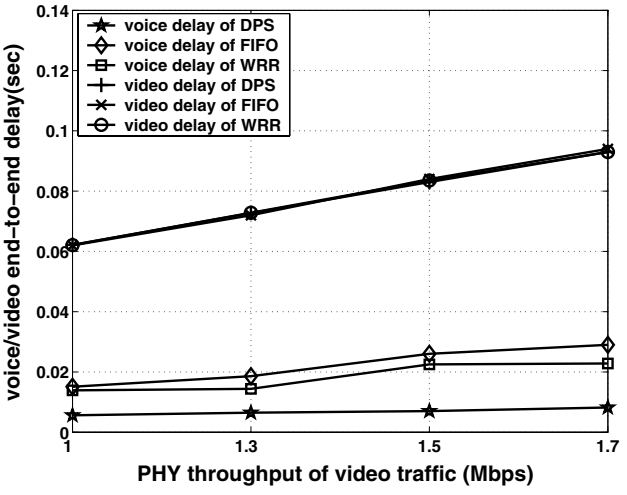


Fig. 7. End-to-end delay of voice and video traffic according to the packet schedulers and the video traffic load

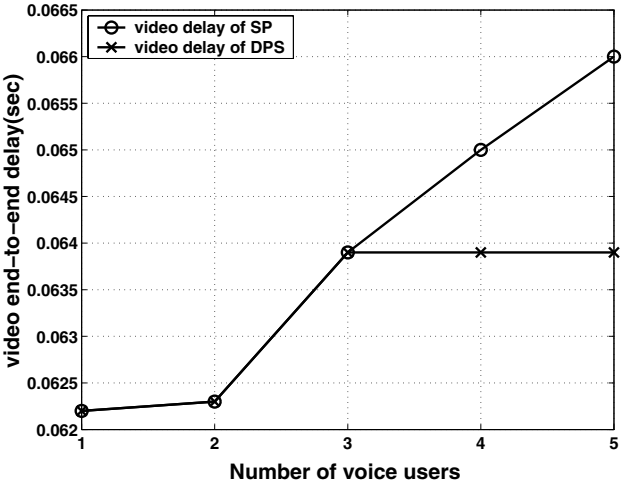


Fig. 8. End-to-end delay of video traffic according to the number of voice users

end-to-end delay of video steaming, FTP and HTTP traffic may increase because of the unfairness of SP. Unlike SP, DPS is able to guarantee the fairness because of the CAC applied to DPS. For example, if we limit the maximum number of voice user in DPS, DPS can drop the voice call when the number of voice users is over the maximum number. Fig. 8 presents the video end-to-end delay when the maximum number of voice users is equal to 3. DPS guarantees the QoS for a video streaming even though the number of voice users rapidly increases.

5 Conclusion

In this paper, we defined the functional features of the network elements and designed the control plane and user plane considering the end-to-end QoS architecture. We proposed the QoS parameter/class mapping and DPS to provide the end-to-end QoS guaranteed service. We also built the simulation model and verified the performance of the proposed technologies. The simulation results show that DPS is superior to FIFO, WRR and SP for the multimedia services such as voice and video.

Since the WLAN and 3GPP interworking technology is the newly embossed issue, our proposed end-to-end QoS mechanisms are expected to be a significant reference. Contributions of this paper are as follows.

- Design the control and user plane
- Propose the QoS parameter/class mapping and DPS
- Build the simulation model using OPNET for the WLAN and 3GPP interworking network

Finally, this framework can be applied to the other heterogeneous network between WiMAX and 3GPP.

References

1. W. Zhuang, Y. S. Gan, K. J. Loh, and K. C. Chua, "Policy-based QoS management architecture in an integrated UMTS and WLAN environment," *IEEE Commun. Mag.*, vol. 41, pp. 118-125, Nov. 2003
2. "QoS and policy aspects of 3GPP - WLAN interworking (Release 7)," 3GPP TR 23.836 v0.4.0, 2005
3. D. Skyrianoglou and N. Passas, "A framework for unified IP QoS support over UMTS and WLANs," *in Proc. European Wireless 2004*, Barcelona, Spain, Feb. 2004
4. S. I. Maniatis, E. G. Nikolouzou, and I. S. Venieris, "QoS issues in the converged 3G wireless and wired networks," *IEEE Commun. Mag.*, vol. 40, pp. 44-53, Aug. 2002
5. "3GPP system architecture evolution : report on technical options and conclusions," 3GPP TR 23.882 v.0.1.1, 2005
6. "3GPP system to WLAN interworking; system description," 3GPP TS 23.234 v6.4.0, 2005
7. "QoS concept and architecture (release 4)," 3GPP TS 23.107 v4.4.0, 2002
8. "Amendment : MAC QoS enhancements," IEEE P802.11e/D13.0, Jan. 2005

Network-Adaptive QoS Routing Using Local Information

Jeongsoo Han

Department of Internet Information technique,
Shingu College, Korea
jshan@shingu.ac.kr

Abstract. In this paper, we propose the localized adaptive QoS routing scheme using POMDP(partially observable Markov Decision Processes) and Exploration Bonus. In order to deal with POMDP problem, we use CEA(Certainty Equivalency Approximation) technique that involves using the mean value of random variables, since it is intractable to determine the optimal solution. And we present a new path selection using the Exploration Bonus method in order to find detour path, which is better than current path and more adaptive to network environment. Also we evaluate performances of service success rate and average hop count which varies with ϕ and κ performance parameters, which are defined as exploration count and intervals.

Keywords: Localized Adaptive QoS Routing, Reinforcement Learning, POMDP, Exploration Bonus, Certainty Equivalency Approximation, Edge-disjoint multi-path.

1 Introduction

In localized routing technique, each node in the network makes local routing decisions based just on local information, i.e., information on the status of the network that has been obtained from the neighbor nodes. Also, routing policies aim to minimize the average overall blocking probability based on local information. If local information is used, it is difficult to determine how a routing decision made on a single node may influence the network's overall performance. It has been demonstrated that localized QoS routing is simple, stable, adaptive and effective in comparison to global QoS routing schemes[1]. [2,3] have proposed a localized QoS routing scheme, proportional sticky routing*(psr)*. The psr scheme maintains self-adaptivity by using the maximum permissible flow blocking parameter and controlling the number of flows routed along a path in each cycle and by re-adjusting flow proportions after every observation period. But, this approach has three main drawbacks. Firstly, the traffic pattern is always unknown in practical networks. Secondly, we need to know the precise information along the path to calculate the blocking probability. Thirdly, even though we are able to identify the traffic pattern and calculate the blocking probability, the iteration time needed to solve the global optimization problem can be significant[4]. To

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 190–199, 2006.
© Springer-Verlag Berlin Heidelberg 2006

overcome the above problems, [4] proposes a reinforcement learning(RL) based scheme(*Q-Learning*) to make path selection in the absence of network traffic pattern and global information in the core networks.

In general, an environment of RL is formalized as MDP(finite-state Markov Decision Process), which calculates the long term values of state action pairs using a temporal difference method for approximating solutions to dynamic programming. But a current optimal action is not guaranteed at future because environments connected with agent can change stochastically over time. In real-world environments, it will not be possible for the agent to have perfect and complete perception of the state of the environment. Unfortunately, complete observability is necessary for learning methods based on MDPs. In this paper, we consider the case in which the agent makes observations of the state of the environments, but these observations may be noisy and provide incomplete information. We will consider extensions to the basic MDP framework for solving partially observable problems. The resulting formal model is called a partially observable Markov Decision Process or POMDP, which starts from a probabilistic characterization of the uncertainty the agent has about the world. Performing dynamic programming to solve a POMDP is highly computationally expensive, and is intractable to determine the optimal solution. Therefor,e we make a form of CEA(Certainty Equivalence Approximation), in which the stochasticity in the world is simplified through the use of expectations. In this paper, we adapt these techniques to routing problems because network environments can change stochastically over time and in absence of global information on each node. We provide a localized routing technique as using POMDP with CEA and a new way of path selection using the exploration bonus method which is more adaptive to network environments. For this, this paper presents the multi-path searching algorithm for the shortest pairs of edge-disjoint paths that revised [10]. Also we evaluate performances of service success rate and average hop count relevant with and performance parameters, which are defined as exploration count and intervals. The organization of this paper is as follows. In chapter 2, we show the proposed routing techniques, updating routing information rules using POMDP with CEA and decision making process with exploration bonus and the multi-path searching algorithm for the shortest pairs of edge-disjoint paths. This is followed by the presentation of performance evaluation in chapter 3. We conclude the paper in chapter 4.

2 Proposed Algorithm

In localized routing technique, each node in the network makes local routing decisions based just on local information. From this viewpoint, it can be connected with POMDP problem which its agent ignores about its environments. That is, that the agent knows a) its own state $x^t(n) \in X$ at all times n within trial t, b) the costs $C_x(a)$ of taking the actions, c) the states that are absorbing, and d) the distribution governing the way the transition probabilities change. But, the agent does not know transition probabilities and its knowledge as to how they change. This ignorance turns a Markov decision problem into a POMDP.

2.1 Routing Model based on POMDP

In order to obtain a proposed network routing model usefully, it is possible to make that the following network's elements correspond to the basic elements of a POMDP : an agent(node which receives a request), the set of actions available to the agent(A, a set of predefined paths to a destination), the state set(X, each network node), a value iteration($V_a^{t,n}(x,y)$, at time n during trial t, the local routing policy which state x for destination node y should be choose). Also, because a source node is ignorant of information about network environments, let $p_{xy}^t(a)$ be a routing success probability(ie, transition probability) when a source node(state $x \in X$) takes a predefined path(action $a \in A$) to destination node(state $y \in X$), where t is the trial number. The network environments are non-stationary in the sense that between routing trials, the transition probabilities can change in a Markovian manner, according to a probability distribution $U[P_{xy}^{t+1}(a)|P_{xy}^t(a)]$(belief state). The agent maintains a distribution over them which it updates based on Bayes's rule in the light of information collected from the environment, uses it for routing decision on state. Therefore, the agent can infer the unknown transition probabilities $p_{xy}^t(a)$ through the distribution over the transition probabilities. And $C_{xy}(a)$ is hop count from state x to destination y when taking action a. Performing dynamic programming to solve a POMDP is highly computationally expensive, and therefore we make a form of CEA, in which the stochasticity in the world is simplified through the use of expectations($q_{xy}^{t,n}(a) = E_{\gamma^{t,n}}[p_{xy}(a)]$) instead of all the $p_{xy}^t(a)$, where $\gamma^{t,n}[p_{xy}(a)]$ is probability distribution at time n during trial t. Fig. 1 shows the network environments using POMDP and CEA techniques in this paper.

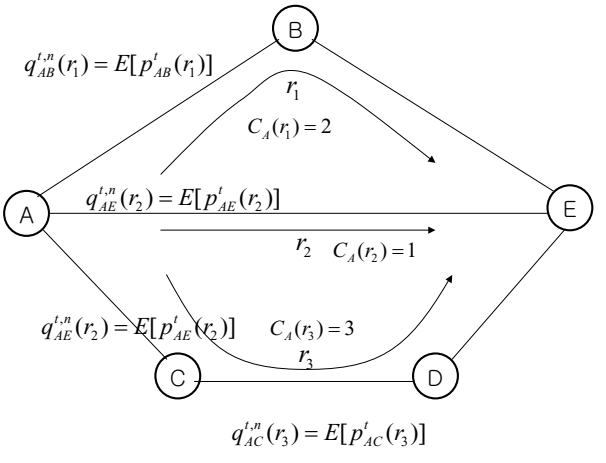


Fig. 1. The network environments using POMDP and CEA techniques

Under the approximation, the agent performs value iteration in the assumed mean process as follows:

$$V_{\alpha+1}^{t,n}(x,y) = \min_{a \in A}\{C_{xy}(a) + \gamma \Sigma_{y \in X} q_{xy}^{t,n}(a) V_{\alpha}^{t,n}(x,y)\} \quad (1)$$

where $0 \leq \gamma \leq 1$ is discount factor and α is the number of the dynamic programaming iteration.

2.2 Updating Routing Information Rules

The course of updating routing information is as follows : Once the agent picks one of the possible actions that minimizes the right-hand side of equation (1), performs it, uses Bayes' rule to compute the posterior distribution over the transition probabilities based on the transition actually observed, and calculates $q_{xy}^{t,n+1}(a)$. The agent attempts to take such an action for admitting a request with certain bandwidth requirement if there is enough bandwidth in the chosen action, so it is possible to route a request on that action. In this paper, we define that such an action is termed as *effective action* and is defined as $e_{xy}^{t,n}(a) = 1$. And if otherwise, *ineffective action* and $e_{xy}^{t,n}(a) = 0$. In this paper, the agent does not know $e_{xy}^{t,n}(a)$, ie it does not know whether it is possible to route on such an action because it does not absent of global information. Therefore, we define a probabilistic model(ϕ) of the efficacy of the transitions ie, the probabilities of routing success on each path. Specifically, let $q_{xy}^{t,n}(a)$be the agent's estimate of the efficacy of action a at x for y at time n during trial t. And the agent assumes that between each trial with some small arrival rate(κ), each $e_{xy}^{t,n}(a)$ gets set to a new probability value, independent of its previous value. After trial $t-1$, the updating routing information at x occurs as follows:

$$q_x^{t,0}(a) = \begin{cases} \kappa\phi + (1-\kappa) q_x^{t-1,0}(a) & \text{(2-1)}, \\ 1 - \kappa(1-\phi) & \text{(2-2)}, \\ \kappa\phi & \text{(2-3)}. \end{cases} \quad (2)$$

When a was not tried at x during the trial, we can define as (2-1). When a was tried at x and was successful, we also define as (2-2). And (2-3) means when a was tried at x and was unsuccessful. $q_{xy}^{t,n}(a)$ is reset when the agent tries a at x to whatever actually happened. For actions that were not attempted, $q_{xy}^{t,n}(a)$ relaxes towards ϕ at a rate governed by the arrival rate κ. And others cases, $q_{xy}^{t,n}(a)$ is reset as each values.

2.3 Decision Making Process(Exploration Bonus)

The decision-making process consists of selecting the path through which the request will be connected. The agent's initial uncertainty about the transition or reward structure of the world should drive the initial experimentation, and, if the world can change stochastically over time, then this further source of uncertainty should drive continuing exploration. Unfortunately, it is computationally intractable for the agent to determine the correct experiments even if it knows what it does not know. And in using this mean process, the approximation fails to directly account for the fact that if a transition on the way to the destination

is blocked because of QoS problem, then the agent will have to choose some other path to that destination. To solve this problem in this paper, we shows the heuristic algorithm, exploration bonus method which is added to the immediate value of a number that is a function of this how long it has been since the agent has tried that action in that state. Therefore, rather than have the agent model the efficacy of transitions, we had it directly add to the immediate cost for a move an exploration bonus of $\alpha\sqrt{n_x(a)}$ for trying action a at state x, if that action had not been tried for $n_x(a)$ trials.

Table 1. Shortest Edge-disjoint Multi-path Searching Algorithm

Initialization]	Let $\delta = \phi$ be the desired path set.
Step 1	[Computation of shortest path] We obtain the shortest path tree T rooted at source s using Dijkstra's algorithm. Let the shortest path from s to destination v be denoted as P_1 and the cost of the shortest path from s to each vertex x be denoted by $C(s, x)$.
Step 2	[Cost transformation and Modification of graph] The cost of every edge (a, b) in G is transformed to $c^{'}(a, b) = c(a, b) + C(s, a) - C(s, b)$. All the edges belonging to T will have a reduced cost equal to 0 due to this transformation. And reverse the orientation of all edges of G which lie in P_1 to form a new graph G_v.
Step 3	[Computation of shortest path] Compute the shortest path from s to v in G_v. Let it be denoted as P_2.
Step 4	[Shortest path pair Generation] The desired path pair, say $P_1^{'}$ and $P_2^{'}$ is obtained by $\{P_1 \cup P_2\} - \{P_1 \cap P_2\}$. And compute $\delta = \{P_1^{'}, P_2^{'}\} + \delta$.
Step 5	[Reduction of graph] We generate new graph $G^{'}$ by deleting all edges belonging to δ from G. And let $G = G^{'}$.
Step 6	[Repeat] We repeat the previous Steps from 1 to 5 until there are no edges connected with s.

2.4 Shortest Edge-Disjoint Multi-path Searching Algorithm

In this paper, we propose a SEMA(Shortest Edge-disjoint Multi-path searching Algorithm) which revises a SSP(Single-Sink Problem) algorithm that S.Banerjee proposed [10] and is used for routing on POMDP routing model This algorithm of finding multiple disjoint path from source to a destination can be formulated as a the minimum-cost flow problem and solved by successive iterations of Dijkstra's shortest path algorithm. Let $G = (V, E)$ be a directed graph having a non-negative cost $c(u, v)$ as associated with each edge $(u, v) \in E$ and let $|V| = n$ and $|E| = m$. The SEMA is as follows Table 1.

If the upper bound of an efficient implementation of Dijkstra's algorithm is denote by $O(n, m)$, SEMA can also be solved in $O(n, m)$ time sequentially.

3 Simulation and Results

This section illustrates how the amount of exploration depends on the performance parameters ϕ and κ using POMDP, CEA algorithms that adapted Exploration Bonus. For this, we will provide the test environments as follows.

3.1 Simulation Environments

As seen in Fig. 2, we use two test network for simulation, and conduct out experiment under the network environment of [1] and the request requirement of [2]. All the links are assumed to be bi-directional and of the same capacity, with C units of bandwidth in each direction. Flows arriving into the network are assumed to require one unit of bandwidth. Hence each link can accommodate at most C flows simultaneously. The flow dynamics of the network are modeled as follows. Flows arrive at a source node according to a Poisson process with rate κ. The destination node of a flow is chosen randomly from the set of all nodes except the source node. The holding time of a flow is exponentially distributed with mean $\frac{1}{\mu}$. The offered network load is given by $\rho = \frac{kNh}{\mu LC}$ where N is the number of source nodes, L is the number of links and h is the mean number of hops per flow, averaged across all source-destination pairs. The parameters used in this simulation are $C = 20$, $h = 2.36$ and $\frac{1}{\mu} = 60$ seconds. We also use multi-paths that generated by SEMA as feasible paths for requests. When we evaluate the Exploration Bonus method, we use the greedy policy until blocking happens and then apply the Exploration Bonus method.

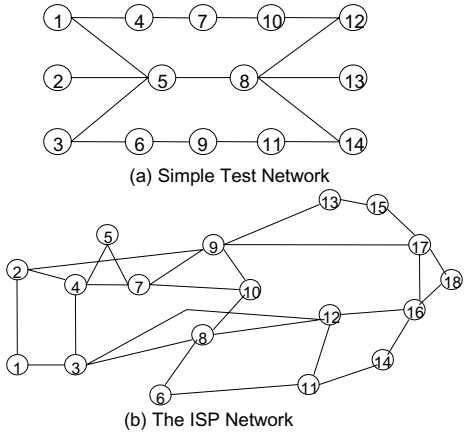


Fig. 2. The Test Networks

3.2 Results Analysis

The agent's experience was divided into a number of routing trials on feasible path. In each trial, the agent starts at the source node and takes actions until it reaches the destination node. In all cases, complete synchronous value iteration was performed before each step in each trial using equation (1) to convert the current model of the world into a suggested action. The value function produced at the previous step was used to initialize value iteration, so convergence took only a few iterations unless the agent found out in the previous step that the world was substantially different from what it expected. The model was updated in the course of a trial as the agent discovered that transitions were effective or ineffective. At the end of a trial, all the expected efficacies $q_x^{t,n}(a)$ were updated to take account of the chance that each transition might have been changed.

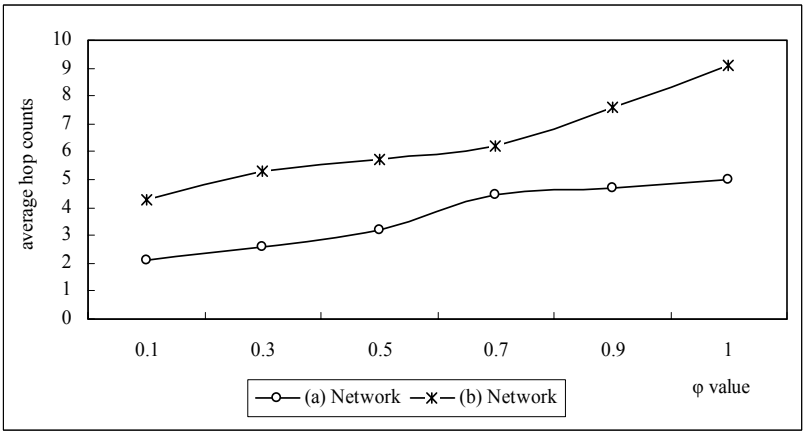


Fig. 3. The performance of routing average hop count on test network which varies with ϕ value

Fig. 3 shows the performance of routing average hop count on each test network which varies with a ϕ value. As ϕ gets bigger, the system will always eventually explore to find paths that might be longer than its current one - ie it will find the detours when the current path will be blocked because of various QoS parameters. But as ϕ gets smaller, the system becomes more pessimistic. It will not bother to find large detours to try possibly better paths. Therefore, the lower ϕ, the more restricted the excursions. As seen in Fig. 3, the actual average length of the shortest paths also increased(from 3.5 hop counts at the real value of $\phi = 0.2$ to 5 hop counts at the real value of $\phi = 0.9$ in the Fig. 2 (a) test network). It is attributed to the fact that it will find detours other than the current path as ϕ increases(κ was fixed at 0.03 and was known to the agent). Also the Fig. 2 (b) test network that has relatively many links between the nodes has more hop counts than the Fig. 2 (a) test network.

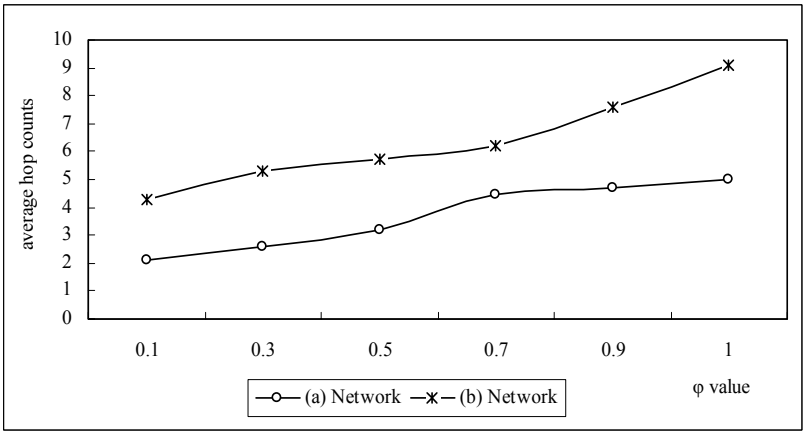


Fig. 4. The performance of routing success rate on test network which varies with ϕ value

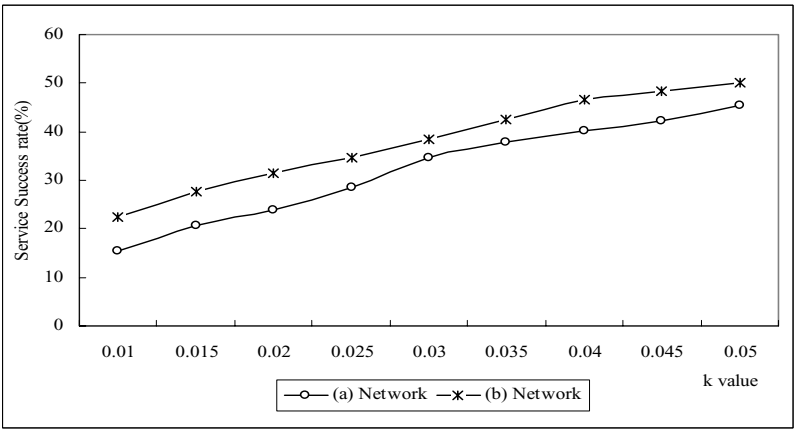


Fig. 5. The performance of routing success rate on test network which varies with κ value

Fig. 4 shows the performance of routing success rate on each test network which varies with a ϕ value. As seen in the result of Fig. 3, the more ϕ increases, the more detours other than the current path the agent explore. Therefore, the performance of routing success rate will be improved. Also κ was fixed at 0.03 and was known to the agent.

Fig. 5 shows the performance of routing success rate on each test network which varies with a κ value. As indicated above, the other parameter κ also has an effect on the relative amounts of exploration and exploitation, but in a more indirect way. Whereas ϕ determines the ultimate amount of exploration about

the current best-known path, κ controls how soon that exploration happens. Therefore, the more frequent explorations, the higher service success rate because it will be clear that increasing κ increased the amount of exploration. ϕ was fixed at 0.5. As seen in Fig. 5, the actual average success rate also increased (from 38% at the real value of $\kappa = 0.03$ to about 50% at the real value of $\kappa = 0.05$ in the Fig. 2 (b) test network). It is attributed to the fact that the amount of exploration will increase.

4 Conclusion

In this paper, we present the new Localized Adaptive QoS routing technology without the knowledge of network global information. For this, we proposed the POMDP model and CEA technology that provide a better QoS routing method using only local information on source nodes and provide a much more adaptive way of path selection depending on a network environment. We also proposed an edge-disjoint multi-path searching algorithm, SEMA. As with POMDPs, we start from a probabilistic characterization of the uncertainty the agent has about the network environment, apply this model to the Localized QoS routing method. And we make a use of CEA, which use the mean values of random variables, because performing dynamic programming to solve a POMDP is highly computationally expensive and is intractable to determine the optimal solution. For better path selection, we propose Exploration Bonus because network environment connected with agent can change over time, so a current optimal action is not guaranteed at future. Also we evaluate performances of service success rate and average hop count which varies with ϕ and κ performance parameters, which are defined as exploration count and intervals. As a result, whereas the parameter ϕ determines the amounts of exploration to find a better path than the current one, κ controls how soon that exploration happens. As ϕ gets bigger, the system will always find the detours when the current path will be blocked because of various QoS parameters. But as ϕ gets smaller, the system becomes more pessimistic. It will not bother to find large detours to try possibly better paths. Therefore, if ϕ gets closer to 1, the success rate and the average hop counts will get higher. Also, the higher κ, the higher service success rate because the amounts of exploration will increase.

References

1. X.Yuan and A.Saifee, "Path Selection Methods for Localized Quality of Service Routing", Technical Report, TR-010801, Dept of Computer Science, Florida State University, July, 2001
2. Srihari Nelakuditi, Zhi-Li Zhang and Rose P.Tsang, "Adaptive Proportional Routing: A Localized QoS Routing Approach", In IEEE Infocom, April 2000.
3. Srihari Nelakuditi, Zhi-Li Zhang, "A Localized Adaptive Proportioning Approach to QoS Routing", IEEE Communications Magazine, June 2002

4. Y.Liu, C.K. Tham and TCK. Hui, "MAPS: A Localized and Distributed Adaptive Path Selection in MPLS Networks"in Proceedings of 2003 IEEE Workshop on High Performance Switching and Routing, Torino, Italy, June 2003, pp.24-28
5. Yvn Tpac Valdivia, Marley M.Vellasco, Marco A. Pacheco "An Adaptive Network Routing Strategy with Temporal Differences", Inteligencia Artificial, Revista Lberoamericana de Inteligencia Aritificial, No 12(2001), pp. 85-91
6. Leslie Pack Kaelbling, Michael L. Littman, Andrew W.Moore, "Reinforcement Learning:A Survey", Journal of Artificial Intelligence Research 4, 1996, pp 237-285
7. P.Marbach, O.Mihatsch, and J.N.Tsitsiklis, "Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming", IEEE Journal on Selected Areas in Communications, Vol. 18, No.2, Feb 2000, pp.197-208
8. Jeong-Soo Han, "A Localized Adaptive QoS Routing using TD(λ) method", Journal of Korean Institute of Communication and Sciences, Vol.30, No.5B, pp. 304-309, 2005
9. Gregory Z. Grudic, Vijay Kumar, "Using Policy Gradient Reinforcement Learning on Automous Robot Controllers", IROS03, Las Vagas, US, October, 2003 [11] Richard S. Sutton etc, "Policy Gradient Methods for Reinforcement Learning with Function Approximation", Advances in Neural Information Processing System, pp. 1057 1063, MIT Press 2000
10. S.Banerjee, R.K. Ghosh and A.P.K Reddy, "Parallel algorithm for shortest pairs of edge-disjoint paths", J.Parallel Distrib. Comput. 33(2):165-171 (1996)

Configuration Management Policy in QoS-Constrained Grid Networks

Hyewon Song[1], Chan-Hyun Youn[1], Chang-Hee Han[1], Youngjoo Han[1], San-Jin Jeong[2], and Jae-Hoon Nah[3]

[1] School of Engineering, Information and Communications University (ICU)
103-6 Munji-dong, Yooseong-gu, Daejeon, 305-714, Korea
{hwsong, chyoun, changhee, yjhan}@icu.ac.kr
[2] Mobile Telecommunication Research Division, Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea
sjjeong@etri.re.kr
[3] Information Security Research Division, Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea
jhnah@etri.re.kr

Abstract. In Grid service, resource management is important to support capability for good quality and efficiency for the computing and storage service. In order to provide this resource management which has low complexity, high performance and high throughput with guaranteeing QoS, the well defined network resource management architecture and scheme have to be considered. Thus, we propose the integrated grid resource management architecture based on QoS-constraint policy and we derive cost-sensitivity policy enforcement procedure which can be applied to this architecture in Grid network. Finally, the results show that the proposed scheme outperforms the conventional scheme in the network performance such as a blocking rate of resource request messages and the operation cost.

1 Introduction

The global growth of Grid computing means the growth in application part reflecting the needs of users. The traditional Grid computing (Computational, Data, Access) [1]-[3] is more specialized, and consist of very various and complex grid services. All the more Resource management need to correspond with the need of users or application and policy based resource management system is one of the most promising candidates for solving requirement of resource management.

For the reliable management of physically distributed and heterogeneous resources, policy-based management [4] has been suggested. The connection between lower network and the resource management system is mandatory for such a grid policy-based resource management system. In other words, it is requisite to make the resource management system abstract because it is hard to manage heterogeneous lower management system managed by local policy of different subject like a grid plane.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 200 – 209, 2006.
© Springer-Verlag Berlin Heidelberg 2006

However, this [4] isn't consider about network resource management of lower network layer such as L3/L2/L1. Actually, since the high performance and throughput and QoS-constraint service requested with a specific Service Level Agreement (SLA) can be supported throughout Grid service networks, the network resource management is also considered importantly. For this network resource management, there are many studies [8]-[10]. Especially, for high throughput issue, optical network is mainly considered. [10] Moreover, as mentioned in [11], optical network with using GMPLS give more efficient management and control scheme for Grid and network resource management in Grid service networks.

In this paper, we propose the integrated grid resource management system architecture based on Quality of Service (QoS) constraint configuration management policy in Grid networks. This QoS-constraint configuration management policy is based on the cost sensitivity using the cost according to providing network resources. Thus, we define the cost function and cost sensitivity function for guaranteeing the QoS in using network resources in this paper, and then, we derive the configuration management policy rule. Also, we propose the policy enforcement process using this policy rule in proposed grid resource management system architecture which is possible to implement in Grid networks. Finally, through the theoretical and experimental analysis, we can show that the proposed scheme outperforms the conventional scheme in network performance and cost performance.

2 Related Works

The Policy Quorum based Resource Management Architecture is a kind of grid resource broker and scheduler that manage resources and jobs in the virtual Grid environment based on Globus Core.[4] Policy quorum, be generated finally, represents the collection of the resources that is adapted by individual policies according to private request. Thus a user is satisfied with QoS by Policy Quorum. And several papers have showed the resource reconfiguration algorithm based on temporal execution time estimation method. Resource reconfiguration performs the reshuffling of the current available resource set for maintaining the quality level of the resources. However there is no consideration of the cost when reallocation job.

The policy-based Grid management middleware sits on the top of Gird Supporting Environment and serves as the first access point for Gird administrator, or software on its behalf, to configure and manage intra-domain and inter-domain Grid resources.

In order to deploy Policy-based management technology in Grid architecture, a policy-based management standard architecture of the Internet Engineering Task Force (IETF) can be used. These policy based management (PBM) system is originated from network management groups to reduce the complexity of management in terms of QoS guarantees. It is suitable to complex management environment such as large scale heterogeneous management in Grid networks. Quorum is a collection of the elements which always intersects. It has been used to maintain the consistency of replicated data, mutual exclusion. In PQRM the intersection property of Quorum is used to make complex resource elements abstracted Grid resource which satisfies various QoS requirements. The main result of PQRM is that QoS-constraint policy

has an advanced method to satisfy QoS requirements requested by applications or services [4].

Although this PQRM has many benefits for Grid resource management, Grid management considered about network resource and service isn't considered like many other Grid resource management architecture. Actually, it is important to consider these network issues in general Grid resource management for supporting better QoS of Grid service. These issues are proposed in GGF, and then, now many researches are going on. [8]-[10] Most of them have an overlay model in a point of architecture view. However, since the integrated architecture is more efficient ultimately, we focus on an integrated model. In this architecture, the GMPLS is supposed for management of network resource in L1/L2/L3. Control and management capability of GMPLS is important to create new services by enhancing the controllability of optical networks. As mentioned in [11], geographically-distributed computing applications for science and enterprise are emerging services for providers, and the optical network is expected to be involved in such the Grid computing applications by using GMPLS protocol. This GMPLS can accept dynamical application specific requirement such as network delay and fault recover time which is requested by geographically-distributed application. Thus, [11] consider a network resource management function block based on management and control scheme in GMPLS networks. Throughout this method, this architecture can get the integrated framework between Grid resource management and network resource management. Similar to this architecture, in this paper, we propose more detailed and modified architecture and procedure in order to manage network resource cost-effectively and consider QoS guaranteed by providing efficient network enforcement scheme based on this architecture. That is, in this paper, we propose the policy based Grid resource management architecture and the cost sensitivity based policy enforcement scheme for the QoS constraint Grid management based on PQRM in proposed architecture.

3 Policy Enforcement Scheme Based on Cost Sensitivity

When a user requests some jobs with a specific SLA, the Grid resource manager select proper resources and reserve selected resources. After that, the job is distributed by a job scheduler and then, is executed in distributed resources. In order to select and reserve proper resources, the policy based resource management scheme using QoS constraint Quorum can be considered in [4]. As mentioned in a previous part, the reservation of network resource can be considered as an important performance factor when the QoS guaranteed grid service is considered. Thus, in this paper, we consider Grid resource management architecture with network enforcement policy based on cost sensitivity referring the PQRM architecture in [4]. We propose the policy enforcement scheme based on cost sensitivity (PES-CS) in this paper. In order to provide our proposed PES-CS, we define the cost sensitivity based network policy enforcement engine (CS-NPEE), which interacts with Grid resource management engine based on QoS constraint Quorum. Namely, throughout this CS-NPEE, the policy determined by Grid resource management engine can enforce to network resources directly. In this process, the proposed PES-CS can be applied in order to select and configure network resources for providing the Grid service when a specific job is

requested. In this section, we define operation cost function and cost sensitivity function. Also, from this function, we derive the policy scheme for the PES-CS and propose the basic procedure for this PES-CS.

3.1 Cost Sensitivity

For network enforcement, we can consider the cost sensitivity based policy enforcement scheme. This scheme is based on network status and cost function when a network node is under a contention because of resource competition. In this section, we define some network cost function, and then, derive cost sensitivity based policy enforcement scheme.

For underlying network, we suppose optical networks based on GMPLS for its control and management process. For this network, we can consider network status such as a status under guaranteeing QoS, NS_{QoS} and a status under contention but guaranteeing tolerable QoS by providing alternative path, NS_{alt}. [5] Using this network status, we can define a function $x(t)$, which means the binary value according to those two statuses, NS_{QoS} and NS_{alt}, as follows.

$$x(t) = \begin{cases} 1, & \text{when } NS = NS_{QoS} \\ 0, & \text{when } NS = NS_{alt} \end{cases} \tag{1}$$

Also, we can derive an operation cost model under various statuses in [5]. Using Eq.(1) and this cost function, the total cost function when providing the service throughout the path between the source *s* and the destination *d* is derived as follows,

$$F_{sd}(t) = x(t) \cdot C_{QoS}^{sd}(t) + (1 - x(t)) \cdot (C_{QoS}^{sd}(t) + C_{alt}(t)) \tag{2}$$

where $C_{QoS}^{sd}(t)$ is a cost based on a Deterministic Effective Bandwidth (DEB) concept in order to guaranteeing QoS by the SLA (Service Level Agreement) of a Grid service requested by clients and $C_{alt}(t)$ is a additional cost of providing a alternate path in order to guarantee the QoS under contention situation in underlying networks [5][6]. This total cost function means the cost in order to provide the path which guarantees the QoS.

When the data for supporting a Grid service is transmitted from source *s* to destination *d* through a network controlled by GMPLS, if the bandwidth for guaranteeing the QoS constraint of this data is assigned, only the cost based on DEB is considered for the total cost function. However, if the bandwidth for guaranteeing the QoS constraint of this data can't be assigned by the reason such as contention of resource or blocking status, the alternate path is needed to guarantee the QoS. Thus, in this case, the total cost function is represented by the sum of the cost based on DEB and the cost resulted from providing the alternate path. Moreover, when it is no meaning that guarantees the QoS because of a continuous increment of the operation cost, the total cost is represented by the penalty cost applied differently according to the service type

because of dropping the data [5]. However, as mentioned in previous section, we assume that the value in the drop status NS_{be} in which the required QoS by SLA is not guaranteed is not considered since our proposed configuration management scheme relates to guarantee the QoS.

When the total cost from Eq. (2) is considered between source and destination, to increase this cost means that the cost for guaranteeing the required QoS increase, especially when the network status changes such as the case of providing the alternate path. When the amount of traffic per each channel is expected by the data scanning in previous section, we can define the sensitivity of the total cost from Eq. (2) as follows:

$$\zeta_F^{sd} = \frac{\partial F_{sd}(t)}{\partial C_{QoS}^{sd}(t)} \tag{3}$$

when the $F_{sd}(t)$ is given by Eq. (2) and $C_{QoS}^{sd}(t)$ means a cost according to DEB cost [5], respectively. ζ_F^{sd} means the variance of total cost according to the variance of the cost based on DEB when traffic flows throughout the path between source *s* and destination *d*.

When the sensitivity of the total cost is given by Eq. (3), we can derive the sensitivity according to the network status using the total cost function, *F*.

$$\zeta_F^{sd} = x(t) + (1 - x(t)) \cdot (1 + \frac{\partial C_{alt}(t)}{\partial C_{QoS}^{sd}(t)}) \tag{4}$$

When we consider the sensitivity according to $C_{QoS}^{sd}(t)$, the sensitivity value of the total cost *F* is dominant to the variation value of both $C_{alt}^{sd}(t)$. Therefore, when the node is under a contention situation, this is, $x(t) = 0$, ζ_F^{sd} dominantly depends on the term, $\Delta = \partial C_{alt}(t) / \partial C_{QoS}^{sd}(t)$, which represents the variation of the cost for providing the alternate path according to the cost based on DEB.

When the alternate path is used at contention situation in Grid networks, the cost for providing this alternate path occurs. This cost increases when the number of hops in provided alternate path increases [5]. In high channel utilization of overall network, selected alternate path includes many hops since the high channel utilization means that most of channels has the traffic load which is closer to the boundary so that most of nodes is under the contention situation. Therefore, the value of Δ can have a positive value because the cost for the alternate path increases. However, if the utilization of channels in overall network is closer the boundary, it becomes hard to reserve the resource for the data. Accordingly, the selected alternate path has to include more hops. This increment of the number of hops causes the increment of the cost. Thus, the value of Δ increases, so that the point in which this value exceeds the upper bound occurs. This upper bound is given by SLA. By this upper boundary, to provide the alternate path is determined. When the sensitivity of total cost, ζ_F^{sd}, has the

boundary by $\zeta_F^{sd} \leq 1+\Delta$, the value exceeding this boundary has no meaning in the network operation cost point of view, so that it needs not to provide the alternate path in this case.

3.2 Policy Enforcement Scheme Based on Cost Sensitivity with PQRM

In this section, we describe the PES-CS procedure and PES-CS algorithm which can apply to Grid networks using proposed Policy based Resource Management Architecture using PQRM.

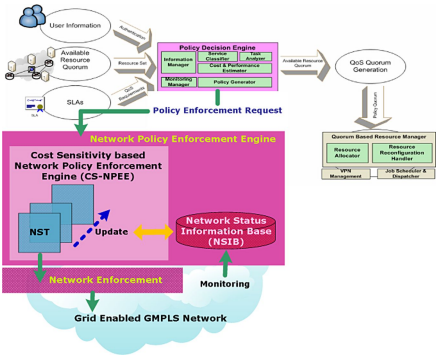


Fig. 1. The basic procedure for policy enforcement scheme based on cost sensitivity with the Policy based Resource Management using PQRM

Fig. 1 shows the basic procedure of this PES-CS for applying to Grid over GMPLS networks. A Network Status Information Base (NSIB) is a kind of data base for collected network status information and a Network Status Table (NST) is an updatable table for selecting network resources. As shown in Fig. 1, the network status information is saved and updated by monitoring procedure in NSIB, and NST in CS-NPEE is created and updated by this information of the NSIB periodically. Moreover, whenever this NST is updated, this updated NST is enforced in that network throughout the CS-NPEE. Moreover, as shown in a top part of Fig. 1, this update and enforcement process interact with QoS Quorum calculated and selected by Policy based Resource Management using PQRM. If the QoS Quorum, which matches with a specific SLA requested by user, is selected by QoS Quorum Generator, this information is reflected in a network enforcement process. Also, during generation of QoS Quorum in QoS Quorum Generator, NST and network status information in NSIB are used in this process like an upper side of Fig. 1.

In the CS-NPEE procedure, the value of NST reflects the network status information which is represented by three network statuses. [5] That is, the value of NST, NS_{ij}, changes according to the network status, and is then updated by the proposed decision algorithm.

When it is assumed that a job with a specific SLA is requested and the contention information in nodes is given for the update of NST, we propose the decision

algorithm in order to provide proper network resources by PES-CS. For these algorithms or procedures, we define some factors in this section. We have the condition factor by the number of hops, Q_h , as follows:

$$Q_h = \begin{cases} 1 & \text{if } H_{sc}^{sd} - H_{cd}^{sd} \geq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

where H_{sc}^{sd} and H_{cd}^{sd} are the number of passed nodes before a current nodes in this path and the number of remaining nodes after a current node in this path, respectively [5]. If the current node is closer to destination *d*, the value of Q_h is one, otherwise, the value of Q_h is zero. Also we can obtain the other condition factor, $Q_{\delta_F^{sd}}$, from the boundary in a previous section and it is defined as follows:

$$Q_{\delta_F^{sd}} = \begin{cases} 1 & \text{if } \zeta_F^{sd} \leq 1 + \Delta \text{ and } x(t) = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

where $\Delta = \partial C_{alt}(t) / \partial C_{DEB}^{sd}(t)$ and $x(t)$ is value according to network statuses. When the decision factor is represented by a sum of above two condition factors, the combined threshold check function can then be stated as

$$\mathrm{C}_{ALT} = \begin{cases} 1 & \text{if } Q_t = w_\delta + w_h \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

Using previous parameters, we can determine the threshold check function. When the current node between source and destination is under a contention situation, if the node that is closer to destination *d* and the value of ζ_F^{sd} , which represents the sensitivity of the total operation cost, is within the tolerable range, the combined threshold check function C_{ALT} is one, so that the node makes a decision to deflect the alternate path. Otherwise, C_{ALT} is zero, so that the node makes a decision to drop the data packet. When information is obtained from NST and NSIB, the threshold check function is determined.

Finally, when the current node is under the contention situation, the node makes a decision whether the data is to be deflected to an alternate path or dropped according to the threshold check function, C_{ALT} . As mentioned in a previous part, the procedure for updating and enforcement in CS-NPEE relates to the upper part – the Policy based Resource Management using PQRM. Fig. 2 shows the flow chart used in CS-NPEE.

This procedure is applied by the CS-NPEE when the job with a specific SLA is requested and the resource scheduling is performed by the Grid resource management engine in the Policy based Resource Management using PQRM.

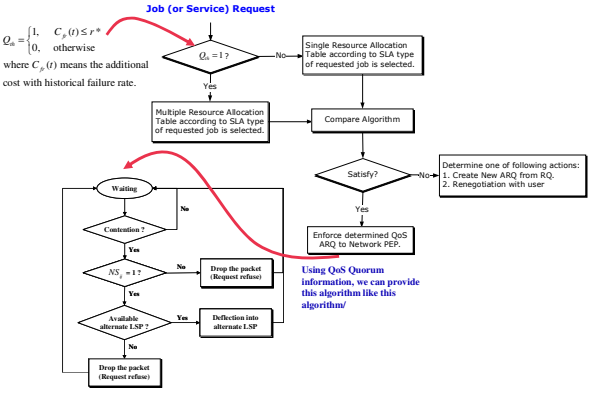


Fig. 2. The Policy Enforcement Scheme based on Cost Sensitivity (PES-CS) procedure with QoS Quorum information by the Policy based Resource Management using PQRM

4 Simulation and Results

In order to evaluate the performance of the proposed policy enforcement scheme, a simulation model is developed. We use the Grid network underlying optical network based on GMPLS in our simulation. The data sources were individually simulated with the on-off model. The simulation is carried out using a 14-node NSFNET network topology. The transmission rate is 10 Gb/s, the switching time is 10 us, and the data header processing time at each node is 2.5 us. The primary paths are computed using the shortest-path routing algorithm, while the alternate paths are the link-disjoint next shortest paths for all node pairs. [5]

Fig 3 shows the results from this simulation. The basic mechanism of proposed PES-CS is different from schemes in general optical network. As mentioned in previous parts, when the node is under the contention situation, the alternate path is provided by threshold value. On the other hand, the conventional general schemes provide retransmitting service at that time. This affects blocking rate in Grid networks. As shown in Fig 3, the blocking rate of the proposed scheme decreases about average 23.679% compared with the general scheme.

Moreover, in order to evaluate the PES-CS scheme in terms of cost, we consider the traffic source that is leaky bucket constrained with an additional constraint in the peak rate based on [7]. We consider 50 different network status tables according to randomly generated traffic patterns under the given conditions. We assume an interval among incoming traffic scenarios is a monitoring interval. For each scenario, we compare the values of total operation cost function between the existing general scheme and the proposed PES-CS. For a comparison, the upper boundary for PES-CS, $1+\Delta$, is assumed to be 100. In the case of existing general scheme, the total cost is about 4 times that of the proposed policy decision in terms of average total cost. This means that existing general scheme provides an alternate path in spite of high cost value. In addition, from the point of view of variation, the cost of existing general

scheme fluctuates widely as shown in (a) of Fig 4. Also, (b) of Fig 4 shows that most of the sensitivity values in the case of PES-CS are constant, at 100.

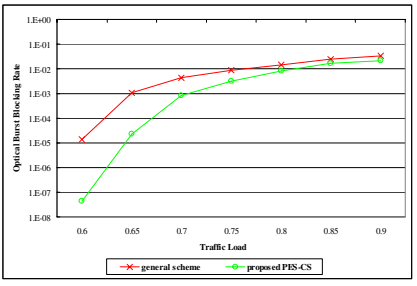


Fig. 3. Blocking rate comparison of general scheme and proposed PES-CS

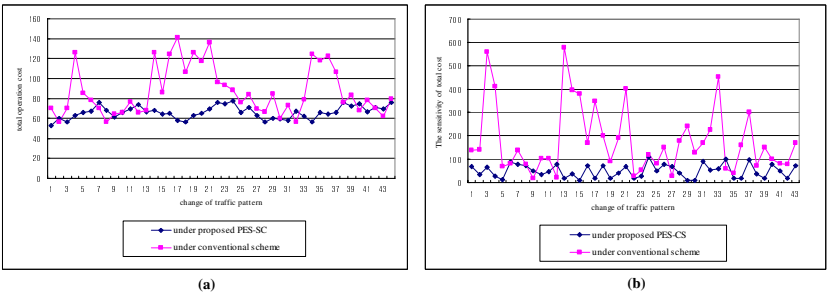


Fig. 4. (a) The total operation cost comparison, (b) The sensitivity comparison

5 Conclusion

In this paper, we proposed policy based Grid resource management architecture based on existing PQRM and procedure in order to manage network resource and consider QoS guaranteed by providing efficient network enforcement scheme based on this architecture. For this policy enforcement scheme for the modified policy based Grid resource management architecture, we developed a modified operation cost model according to the network status information changed by guaranteed QoS. In addition, using the bounded range of the sensitivity of this cost, we proposed a network status decision algorithm, and developed policy decision criteria for policy enforcement in Grid networks by providing an alternate path. As shown in the comparison of the cost performance between our proposed scheme and conventional schemes, our scheme is performed under a stable state. As well, in comparing the blocking rate between our proposed scheme and conventional schemes, ours has good performance in terms of blocking rate. Finally, by using the bounded range of the sensitivity of the total operation cost, our proposed scheme has a reducing effect of about 24% in terms of total operation cost.

Acknowledgement

This research was supported in part by ITRC (Information Technology Research Center) and MIC (Ministry of Information and Communication) of Korean government.

References

1. Foster, I. and Kesselman, C. (eds.). "The Grid: Blueprint for a New Computing Infrastructure". Morgan Kaufmann, 1999
2. Foster, I. and Kesselman, C. "The Anatomy of the Grid:Enabling Scalable Virtual Organizations". Intl J. Supercomputer Applications, 2001
3. Czajkowski, K. et al. ". Grid Information Services for Distributed Resource Sharing, 2001
4. Byung Sang Kim et al. "Policy Quorum based Resource Management Architecture in Grids", IJCSNS 2005, Vol 5, No 8
5. H.W. Song, S.I. Lee, C.H. Youn, "Configuration Management Policy based on DEB Cost Model in OBS Networks", ICAS-ICNS 2005, Oct. 2005.
6. C.H. Youn, H.W. Song, J.E. Keum, L. Zhang, B.H. Lee and E.B. Shim, "A Shared Buffer Constrained Topology Reconfiguration Scheme in Wavelength Routed Networks" INFORMATION 2004, Nov. 2004.
7. D. Banerjee and B. Mukherjee, "Wavelength routed Optical Networks: Linear Formulation, Resource Budget Tradeoffs and a Reconfiguration Study," IEEE/ACM Transactions on Networking, Oct. 2000.
8. Masum Z. Hasan, et al, "Network Service Interfaces to Grid", draft-ggf-masum-grid-network-0, GGF Draft, May 2004.
9. Doan B. Hoang, et al, "Grid Network Services", draft-ggf-ghph-netwerv-2, GGF Draft, May 2005.
10. D. Simeonidou, R. Nejabati, et al, "Optical Network Infrastructure for Grid,", draft-ggf-ghpn-opticalnets-1, GGF Draft, Mar. 2004
11. M. Hayashi, T. Miyamoto, T. Otani, H. Tanaka, A. Takefusa, et al, "Managing and controlling GMPLS network resources for Grid applications," OFC 2006

A Proposal of Requirement Definition Method with Patterns for Element / Network Management

Masataka Sato, Masateru Inoue, Takashi Inoue, and Tetsuya Yamamura

Nippon Telegraph and Telephone Corporation
Access Network Service Systems Laboratories,
NTT Makuhari Bldg. 1-6 Nakase Mihama-ku Chiba-shi, Chiba 261-0023 Japan
{sato.masataka, inoue.masateru, inoue.takashi, yamamura}@ansl.ntt.co.jp

Abstract. For the efficient development of Operation Support Systems (OSSs), we propose a requirement definition method to help specifiers create high quality specifications easily and quickly. A feature of the method is that it uses requirement definition patterns based on common knowledge. The patterns provide solutions to problems in the definition process and help specifiers. We construct requirement definition patterns for Element / Network Management Systems, such as a pattern with effectiveness to unify sequence diagrams related to system behavior. The method also includes a mechanism to help anyone to use the patterns easily that is characterized in using roles in sequence diagrams. To verify the effectiveness of our proposal, we use two case studies to show that our method has reduced the time needed to create specifications, and has improved their quality.

1 Introduction

Telecommunication service providers are using Operation Support Systems (OSSs) to realize quick and appropriate provision of, for example, broadband services. Recently, acute competition between providers has led to a demand for a more effective way to develop OSSs. However, OSS development is becoming more complex, costly and time consuming.

Therefore, many technologies are being studied with a view to improving OSS development. For example, source code generation from models such as the Model Driven Architecture (MDA) [1] is helping to improve implementation productivity. However, OSS development is not always improved even if the implementation productivity is improved. This is because the required development and the actual development might differ. If there is a difference, the implementation is repeated. Additionally, research shows that incorrect, missing and unclear requirement specifications lead to a post-process overrun in terms of cost and time [2]. This implies that the overall improvement of OSS development is not achieved solely by an improvement in implementation productivity. Additionally, because the cost of rectifying errors in implementation corresponds

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 210–220, 2006.
© Springer-Verlag Berlin Heidelberg 2006

to 10-20 times the cost of rectifying errors in the requirement definition [3], it is possible to improve the overall OSS development by improving requirement definition. We therefore focus on improving the requirement definition process in order to reduce the cost and time of OSS development.

We analyze the requirement definition process to enable us to improve it as described below. In general, it is difficult to elicit complex requirement of OSSs and create a requirement specification precisely. Consequently, an inexperienced specifier creates low quality specifications that include errors, omissions and lack of clarity. We believe that if the differences between knowledge possessed by specifiers are reduced and every specifier has a high level of skill based on that knowledge, the specifications will be of high quality. For that purpose, we patternize some common knowledge used for creating specifications. Specifiers refer to the patterns and can find certain solutions to problems in the requirement definition phase.

Recently, software patterns have been used in various software development phases, such as software design [4], and software architecture [5]. We apply such patterns to the requirement definition phase. In fact, four patterns were introduced in the requirement definition phase and their effectiveness was reported [6]. Moreover, technologies for using patterns are required. For example, certain development tools [7] have a function for defining patterns, and applying them to design specifications or source codes. However, these tools don't have functions for the requirement definition patterns.

We propose a mechanism to facilitate pattern use in the requirement definition phases. The mechanism has functions to help specifiers select appropriate patterns and use them unconsciously. A feature of the mechanism is the use of roles in sequence diagrams. We aim to improve OSS development through the requirement definition patterns and the related mechanism.

In this paper, we propose a requirement definition method that includes the above patterns and mechanism. That is, a specifier of any skill level who uses the method will be able to create high quality specifications easily and quickly. First, we outline the problems related to the requirement definition phase. Then, we detail a method for solving these problems by using patterns. After that, we describe a mechanism that assists anyone to use the patterns easily. Finally, we evaluate our proposed method using two case studies.

2 Problems of Requirement Definition

We describe the problems related to the requirement definition phase and why requirement specifications are difficult to create.

If specifications include errors and these errors are discovered in later phases, specifiers must rectify them. Moreover, if specifications are lacking, the specifiers must redefine the requirement by returning to the requirement definition phase. Essentially, rectifying and redefining are unnecessary and require extra cost and time.

In addition, even if there are no errors or omissions, there is another problem, namely, the specifications content differs depending on the specifier or the development project. The differences related to the nature of the content, the notation that is used, and the specifiers' policy. These differences lead to the specifications becoming unclear. Therefore, it may take another person a long time to understand them and misunderstanding may occur. This also results in additional cost and time.

To overcome these problems, we need a requirement definition method that helps specifiers to avoid errors omissions and lack of clarity. Certain methods have been developed to achieve these goals. For example, the standard content that should be included has been defined [8]. In the Unified Process [9], a requirement definition process is constructed, which includes procedures, roles, and viewpoints. However, these methods are not employed by many software development projects. We believe that one of the reasons for this is that they are unsuitable for the developed OSS or the scale or the development project. As a result, the proposed method must be customized by each project and so is unlikely to be employed.

Therefore, we propose a method specified for OSSs of a particular domain, and solve the problem posed by the general methods. That is, the method specified for OSSs of a particular domain does not need to be customized according to the project. In this paper, we focus on OSSs for element / network management.

To develop a method specified for the OSSs, we analyze a feature of the OSSs. To achieve the objective of managing network elements (NE) such as equipment, the OSSs must register the network element as a managed element. By necessity, any management OSS has a function for registering elements. Similarly, to achieve the objective of managing a network, the OSS also registers network elements, such as equipments or other OSSs. And, this function is classified under Telecommunications Management Network (TMN) in ITU-T [10] and is called Configuration Management.

In consideration of the above-mentioned feature, we assume that there is certain common knowledge available for defining the element / network management requirement. This knowledge provides what the specifiers should define and how they should define it. Usually, specifiers obtain such knowledge through experience of the requirement definition process. Therefore, this knowledge is different for each person and the results of requirement definition are therefore also different.

In consequence, we believe that the different degrees of knowledge possessed by the specifiers are a problem in terms of the requirement definition phase.

3 Overall Image of Proposed Method

We show the overall image of proposed method.

To reduce the difference between the specifiers' knowledge levels, we employ a method for sharing common knowledge. By sharing common knowledge,

– It is possible to reduce knowledge-include differences between requirement definition results. This implies that we can suppress any lack of clarity as regards the specification.
– It is possible to enhance knowledge of immature specifiers. This implies that errors and the omissions of specifications will be prevented and the time needed to think about unknown topics will be reduced.

In order to share common knowledge, we patternize the knowledge used for creating specifications. Generally, a pattern in software development is defined as follows. "A software pattern is a solution to the problem that appears repeatedly in each phase of the software development". In this work, we construct requirement definition patterns based on common knowledge, and the patterns provide specifiers with solutions to create specifications. Specifiers will share this common knowledge by using patterns unconsciously.

Moreover, technologies are required that enable us to use patterns because it is difficult to select appropriate patterns. So, we propose a mechanism designed to help anyone select appropriate patterns and use them easily. If the mechanism does not exist, specifiers cannot completely share common knowledge and there are differences in pattern usage according to the specifiers' skill.

In summary, our proposed method includes the following two features.

– The use of patterns based on common knowledge.
– A mechanism to help anyone select appropriate patterns and use them easily.

We detail our requirement definition process in section 4 and show our constructed patterns for element / network management in section 5. We describe about the mechanism in section 6.

4 Defining Requirement Definition Process

First, we define the requirement definition process and artifacts created as a result of that process. Note that there are two types of requirement, a functional requirement and a nonfunctional requirement, and we do not deal with the latter.

Our requirement definition process demands the following artifacts:

(A) Business Flow
(B) Usecase Diagram
(C) Data Analysis Model
(D) Sequence Diagram

The artifacts are created in the following flows.

1. A business task flow is described as a Business Flow.
2. For the each business task in the business flow, the specifier extracts a function that operators operate. Then, a Usecase Diagram is described that aggregates extracted usecases.
3. The specifier extracts OSS's management data to achieve usecases and input / output through a screen. Then, a Data Analysis Model is described that aggregates extracted data.

4. The specifier describes the behavior of the OSSs to realize each usecase as a Sequence Diagram.

And, the relationships between the artifacts are as follows (Fig 1).

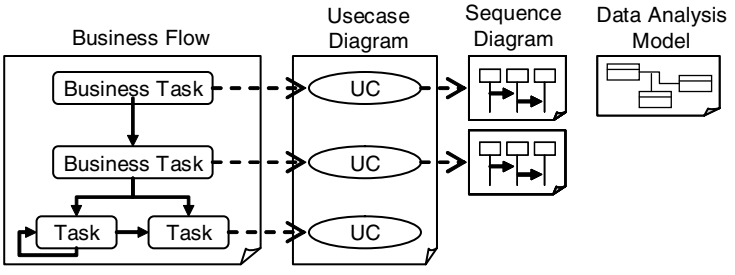


Fig. 1. Relationships between artifacts

It should be remembered that the specifier does not always create these diagrams with the above procedure. In addition, the diagrams are modified repeatedly after their creation and it is necessary to maintain their mutual consistency.

5 Patterns for Element / Network Management

We define the requirement definition patterns for element / network management. As shown in section 3, we first patternize the common knowledge related to creating specifications. Specifiers refer to the patterns and use them. Pattern use means to share common knowledge unconsciously.

A pattern based on common knowledge includes a solution to a problem that appears repeatedly while creating artifacts for the requirement definition phase. An example of the solution is as follows. When a specifier encounters such problems as "How do I divide it?" "What structure is suitable?" "In what order should I use them?", a pattern provides a solution to the problem such as "This is the size" "Connect A and B" "The order is 1234". By obtaining these solutions, the specifiers reduce the time they need to think about problems and unify the specifications and know what should be done even if their skill level is not high. As a result, using patterns helps specifiers to create high quality specifications easily and quickly.

In addition, in this paper a pattern consists of **a problem**, **a solution**, **an explanation** and **effectiveness**. The problem describes what specifiers will try to do and the solution instructs the specifier as to how to solve the problem. For the purpose of defining patterns, we investigate the requirement specifications of six OSSs constructed in our division and construct 25 requirement definition patterns.

One of these patterns is as follows.

Problem. When an OSS registers certain management data to a database and a NE, the OSS has a possibility of both registering it to a DB after a NE and registering it to a NE after a DB. When both orders of registering to a DB and a NE exist in the specification, the specification loses unity. If a registering order is changed depending on diagrams, it is hard to understand the semantics of the diagram

Solution. First, the OSS registers to the DB. Then, the OSS registers to the NE.

Explanation. The order of access to the DB and access to the NE is interchangeable. The OSS can access either first but the order should not be changed in the OSS. So, we decided that the DB access is first.

Effectiveness. The pattern unifies the order and eliminates the lack of semantic clarity.

An example of a sequence diagram to which the pattern was applied is shown in Fig. 2.

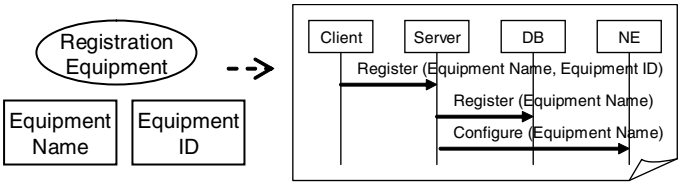


Fig. 2. Example of a sequence

We defined other patterns in relation to creating artifacts but the details are omitted here. The patterns are also effective for such purpose as reducing creation time, and unifying diagrams.

6 Mechanism for Facilitating Pattern Use

As described in section 3, we propose a mechanism to help anyone select appropriate patterns and use them easily. If the mechanism does not exist, there are differences in pattern usage according to the specifiers' skill. This mechanism is an element of our proposed method.

In particular, we think sequence diagrams are more difficult to create because they are more complex than other diagrams. Here, we propose the Join Point concept, which is described in the next section, as a mechanism that facilitates pattern use. Join Point is effective in the following ways.

- Join Points reveal what the specifier should do and think about in terms of using patterns. So, Join Points help any specifier to create sequence diagrams easily.
- Sequence diagrams that satisfy patterns can be created.

6.1 Analysis of Sequence Diagrams

Figure 3 shows a sequence diagram describing "register equipment".

First, we notice that the meaning is different although the same "Equipment ID" is used depending on the message. For example, "Equipment ID" in the message "1. Register (Equipment Name, Equipment ID, IP Address)" means data are input through a client. On the other hand, "Equipment ID" in message 2 indicates the search condition of the DB and "Equipment ID" in message 5 means the configuration data of the equipment.

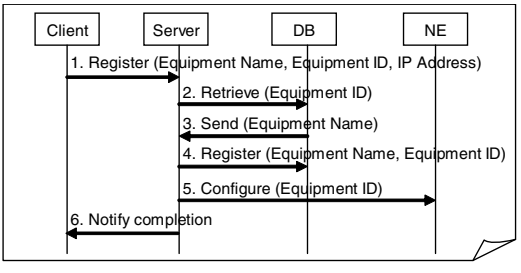


Fig. 3. Example of "register equipment" sequence diagram

Therefore, we think that the management data in OSSs have various roles in a sequence diagram. For example, a role in the message 1 is "User Input", the role in message 2 is "Search condition of DB", the role in the message 5 is "Configuration data for Equipment". We think that such roles are common in element / network management OSSs.

In the following section we detail the mechanism to help anyone to employ the method by using such roles

6.2 Using Roles in Sequence Diagrams

The management data roles are common to all OSSs but each OSS has certain differences. The differences consist of which management data have roles. Figure 4 shows an example. "Equipment ID" has a role of "Configuration data for Equipment" in System A while "Equipment No" has the role in System B.

In other words, a sequence diagram means a relationship in which management data have roles. We named the role "Join Point" and a specifier makes a new diagram by changing the relationship.

The procedure is as follows. We extract the roles in sequence diagram as Join Point in advance and define the notation and structure of sequence diagram related to Join Point. Then, we use the Join Point to generate sequence diagrams. For example, the role "Configuration data for Equipment" generates a message whose name is "configure ( )" from Server to NE. At this time, the generated messages in the diagrams are defined to satisfy the notations and structures defined by patterns. That is, specifiers can use various words (e.g. configure, set,

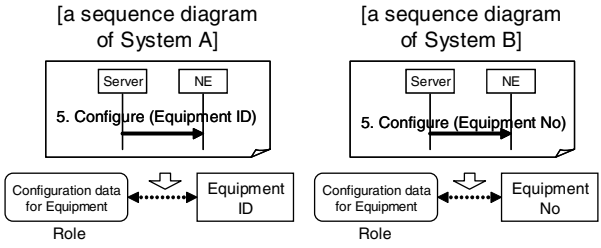


Fig. 4. Differences in roles

register, increase) as "to configure equipments" but patterns define only the use "configure". Additionally, it concerns the word as well as the structure.

In addition, a sequence diagram is generated by not only a single Join Point but also relationships between multiple Join Points. In a simple example, a messages order is defined by the relationships. That is, when an OSS obtains data from a DB, there are two Join Points "search condition of DB" and "taking data from DB". These Join Points have a relationship whereby "search condition of DB" comes earlier than "taking data from DB".

Additionally, these Join Points generate some elements of a sequence diagram and a frame of sequence diagrams can be constructed by collecting them. The frame is not a complete sequence diagram and has some blanks that mean it is unknown which management data have roles in Join Points (Fig.5).

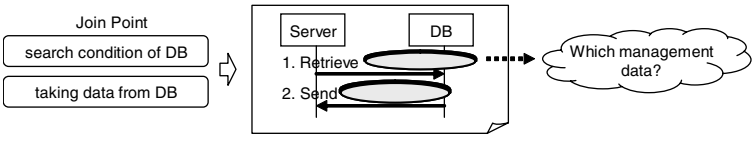


Fig. 5. An example of a frame

By associating management data with a Join Point, the management data have roles in the Join Point and the blanks are filled. Then, the frame becomes a complete sequence diagrams. That is, a sequence diagram is generated by the relationship between Join Points and management data automatically after associating management data with Join Points.

In consequence of this automatic generation, specifiers need not create diagrams and only have to associate. (Fig. 6)

Even if specifiers do not know OSSs well and are inexperienced, they can create sequence diagrams satisfying patterns easily and quickly by using automatic generation.

In summary, Join Points provide the following effectiveness.

– Specifiers do not have to select patterns because Join Points include them in advance.

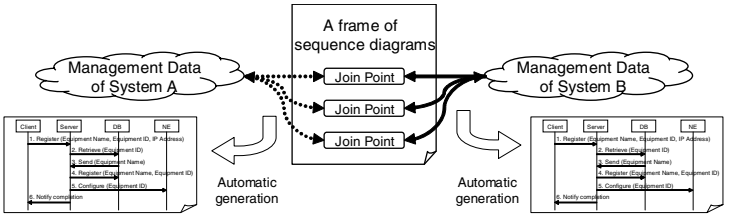


Fig. 6. Creating sequence diagrams by Join Point

- Join Points conceal the existence of sequence diagrams from specifiers because a sequence diagram is generated automatically.
- Sequence diagrams that satisfy patterns are generated.

As a result, specifiers can create sequence diagrams satisfying patterns easily and quickly.

7 Case Study

Next, we evaluate our proposed method by applying it to the development of two OSSs, specifically to make sequence diagrams by using Join Point. One of the OSSs manages layer 2 networks and the other manages ATM networks. The evaluation points are time reduction, quality of sequence diagrams such as unity, and whether or not the method helps specifiers to make sequence diagrams.

Prior to the evaluation, we made a tool that generates sequence diagrams by using Join Point. The tool is capable of reading relationships between management data and Join Point and generates sequence diagrams with an XML Metadata Interchange (XMI) format.

7.1 Outline

We assume that two OSSs are developed by two specifiers who have the same level skill and they create sequence diagrams related to the OSSs. Specifier A uses our proposed method (patterns and Join Points) and specifier B uses the conventional method to create the diagrams by himself. After both specifiers finish the creation, we measure the time until the sequence diagrams are created and their quality. Finally, we compare the time and quality, and consider whether or not the proposed method help in the creation process.

The two specifiers create under two different conditions. One is to create new diagrams related to OSS C. We call this case 1. The other is to change existing diagrams to adapt to other OSS D. We call this case 2.

7.2 Result

In case 1, specifier A took 55 minutes to create five sequence diagrams while specifier B took 93 minutes. In case 2, specifier A took 28 minutes while specifier

B took 55 minutes. In both cases, specifier A created diagrams more quickly than specifier B. Moreover, the quality of the sequence diagrams created by specifier B was poorer than that of the diagrams created by specifier A in both cases. This was because the words used in the diagrams were varied in some places. For example, different words were used for the same meaning, such as "access to NE" and "get from NE". Additionally, the diagram structures were not uniform, for example the order of messages was different in each diagram. The two case studies revealed that our method reduced the time and improved the quality of diagram creation.

In addition, we wanted to evaluate whether or not the method helps specifiers to create sequence diagrams. In order to evaluate this, we considered the specifiers' impressions to be major indicators and we collected their comments. For example, specifier A said that he could create and change diagrams by considering the management data that relate Join Points, and that the approach was easier than the conventional method. This comment indicates that our method helped him to create sequence diagrams.

	Case Study 1			Case Study 2		
	Time (min)	Quality	Easy to create	Time (min)	Quality	Easy to change
Proposed Method	55	Good	Good	28	Good	Good
Conventional Method	93	Average	Average	55	Average	Average

Fig. 7. Results of two case studies

According to these two case studies, we got a result our method reduces time to create sequence diagrams and improves quality of them and helps specifiers.

8 Summary and Future Work

With a view to developing OSSs more efficiently, we proposed a requirement definition method to help specifiers to create high quality specifications easily and quickly. A feature of the method is its use of requirement definition patterns based on common knowledge. The patterns provide solutions to problems in the definition process and help specifiers. Additionally, our proposed method includes a mechanism to assist anyone to create sequence diagrams easily, characterized in using Join Points which indicate roles in sequence diagrams. When specifiers use Join Points, they can know what they should do and think about in terms of pattern use, and they can create sequence diagrams that satisfy the patterns. We used two case studies to show that our method reduces the time required to create specifications and improves their quality. A specifier that used our proposed method commented that he was able to create specifications by using patterns and the proposed mechanism. In the future, we will define exhaustive patterns to create requirement specifications and verify their effectiveness.

References

1. Frankel, D.: Model Driven Architecture. John Wiley & Sons (2003)
2. The Standish Group: The CHAOS Report (1994)
3. Boehm, B.W.: Software Engineering Economics. Prentice Hall (1981)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
5. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns. John Wiley & Sons (1996)
6. Hagge, L., Lappe, K.: Sharing requirements engineering experience using patterns. IEEE Software **22**(1) (2005) 24–31
7. Borland Software Corporation: Together. (http://www.borland.com/)
8. IEEE: IEEE 830 Documentation Standard for a Software Requirements Specification. (1998)
9. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley (1999)
10. ITU-T: ITU-T Recommendation M.3010 (2000)

Distributed Fault Management in WBEM-Based Inter-AS TE for QoS Guaranteed DiffServ-over–MPLS*

Abdurakhmon Abdurakhmanov, Shahnaza Tursunova, Shanmugham Sundaram, and Young-Tak Kim**

Dept. of Information and Communication Engineering,
Graduate School, Yeungnam University
214-1, Dae-Dong, Kyungsan-Si, Kyungbook, 712-749, Korea
graf_best@yahoo.com, saturn_1986@mail.ru,
shanmughams@gmail.com, ytkim@yu.ac.kr

Abstract. Distributed fault management and event notification are essential in Inter-AS Traffic Engineering (TE). In this paper we design and implement distributed fault management for WBEM based inter-AS TE. We designed DMTF Managed Object Format (MOF) based Managed Object (MO) for fault management. Using the Event Model in standard Common Information Model (CIM) in DMTF WBEM we implemented fault indication and SNMP trap handling mechanisms in indication and SNMP providers accordingly on OpenPegasus [1]. We extended existing providers with new SNMP provider which is not supported by OpenPegasus WBEM, and SNMP manager functionalities. The detailed design and implementation of SNMP, indication, instance and other C++ based providers on OpenPegasus WBEM for inter-AS TE are explained.

1 Introduction

Recently, traffic engineering has been increasingly emphasized in end-to-end QoS guaranteed multimedia service provisioning in next generation Internet. MPLS can provide efficient traffic engineering by configuration of Traffic Engineering - Label Switched Path (TE_LSP) among MPLS Label Switched Routers (LSR) [2]. In traditional IP network, the failure in link/node has been restored after long delay from the fault occurrence by Link State Advertisement (LSA) of routing protocol, which is notified by flooding mechanism among adjacent routers, where each router changes routing table to by-pass the erroneous link or node. In MPLS network, unlike the connectionless IP network, we can achieve the high restoration performance using protection switching function that establishes backup LSP for working LSP dynamically or explicitly.

In 1:1 or 1+1 path protection switching scheme, the backup path should be pre-established to provide the fast protection switching performance. However, the backup LSP for fast restoration is not used in normal traffic transmission until any

* This research was supported by the MIC, under the ITRC support program supervised by the IITA.

** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 221 – 231, 2006.
© Springer-Verlag Berlin Heidelberg 2006

LSP reduces the link utilization. The primary goals of fast restoration by fault management function are (i) fast restoration of quality of service (QoS) guaranteed differentiated path, and (ii) the guaranteed bandwidth of backup LSP at fault occurrence.

In this paper, we design and implement distributed fault management in WBEM based inter-AS TE for QoS guaranteed DiffServ-over-MPLS service provisioning**.** We extend WBEM server with new functionality such as fault manager and SNMP provider. SNMP provider includes SNMP trap handler module which receives SNMP trap from the network device and fault manager to provide distributed fault management in inter_domain DiffServ-over-MPLS network. We also propose the design and implementation of MOF based MO for fault management in WBEM-based inter-AS TE.

The rest of this paper is organized as follows. In section II, the related work on WBEM-based CIM-MOF for fault management and SNMP trap notification are briefly introduced. In section III the proposed functional architecture for WBEM-based fault management is explained with fault protection and restoration schemes. In section IV we explain detailed implementation of distributed fault mananagement in WBEM-based inter-AS TE and the performance analysis. Finally, section V concludes the paper.

2 Related Work

2.1 WBEM CIM-MOF for Fault Management

The DMTF has developed a core set of standards that makes up WBEM, which includes the Common Information Model, CIM-XML, CIM Query Language, WBEM Discovery using Service Location Protocol (SLP) and WBEM Universal Resource Identifier (URI) mapping [3]. CIM Core Model has CIM Event Model, which contains a mechanism to precisely identify the occurrence of the phenomenon of interests. This model has been implemented in OpenPegasus, which is an Open Source Software (OSS) implementation of the DMTF WBEM standard.

Fig.1 depicts OpenPegasus indication mechanism. Indication provider and indication consumer should register on CIM server and CIM listener, respectively. Then CIM client creates Subscription to CIM server which loads the Indication provider. From this time, when an event occurs, the indication provider creates an instance of the indication.

Then CIM Server checks the filters in order to determine which one allows it through and finds those handlers which are interested in this event. The indication consumer receives the information about indication from that handler.

The MOF for Indication hierarchy is shown in Fig. 2, and is used to describe the type of events that can be detected. Here CIM_Indication is an abstract class and serves as the base class for all Indication classes. This hierarchy is extensible, so new classes can be added to capture vendor-specific properties and event types. In this hierarchy the subclasses denote the occurrence of an action on an instance, on a class definition and a phenomenon of interests.

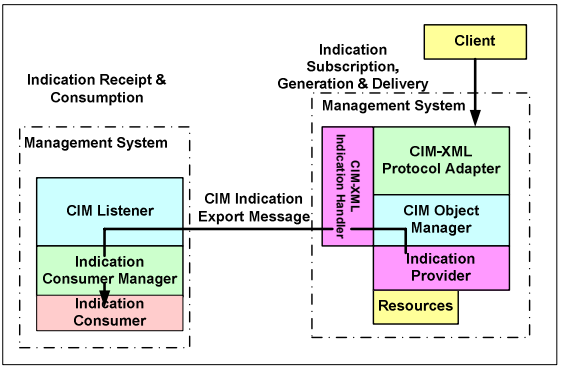


Fig. 1. The fault indication mechanism

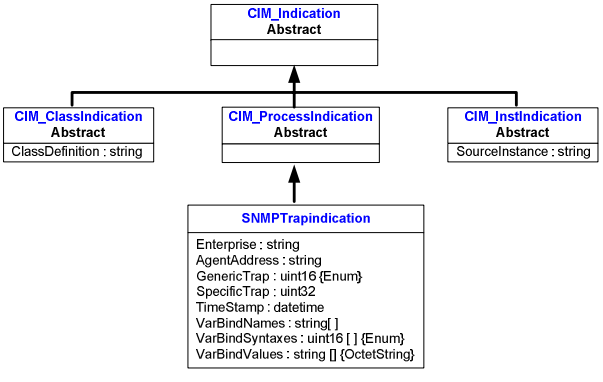


Fig. 2. MOF for fault indication

An indication filter can contain condition or property list of fault notifications to sort indications and if created indication is not satisfied by the any filters, it can not be delivered, and just deleted. If indication filter does not have any query or condition to sort indications, every indication will be delivered to all handlers which are subscribed with that Filter. The format of this query in the filter depends on the language chosen to express it. OpenPegasus supports Web Query Language (WQL) and CIM Query Language (CQL).

Table 1. MPLS Traps

#	Trap	Alarm Description
1	LdpSessionUp	LDP session up between <LdpEntityLdpId> <LdpEntityIndex> <LdpPeerLdpId>
2	LdpSessionDown	LDP session down between <LdpEntityLdpId> <LdpEntityIndex> <LdpPeerLdpId>
3	VrfIfUp	Interface <ifIndex> associated with VRF <VpnVrfName> is up
4	VrfIfDown	Interface <ifIndex> associated with VRF <VpnVrfName> is down
5	TunnelUp	Tunnel <TunnelIndex> up
6	TunnelDown	Tunnel <TunnelIndex> down
7	TunnelRerouted	Tunnel <TunnelIndex> rerouted

2.2 SNMP Trap Notification

A SNMP trap [4] is basically asynchronous notification set from an SNMP agent to a network management system. Like other messages in SNMP, traps are sent using UDP. A trap is a bundle of data that's defined by a managed information base (MIB). Trap falls into two categories: generic and enterprise-specific. We are more interested on MPLS layer specific traps supported by Cisco routers, as shown in Table 1.

3 Architecture of WBEM-Based Fault Management

3.1 WBEM Based Distributed Fault Management Functional Architecture

Fig.3 depicts WBEM based distributed fault management architecture for inter-AS TE. In addition to traditional WBEM main components [5], there are several new extended modules such as Fault Manager. SNMP provider is used for fault management. The upper part of Fig.4 is WBEM Server (i.e., NMS) and lower side is a real network element (NE) that includes nodes such as IP/MPLS router.

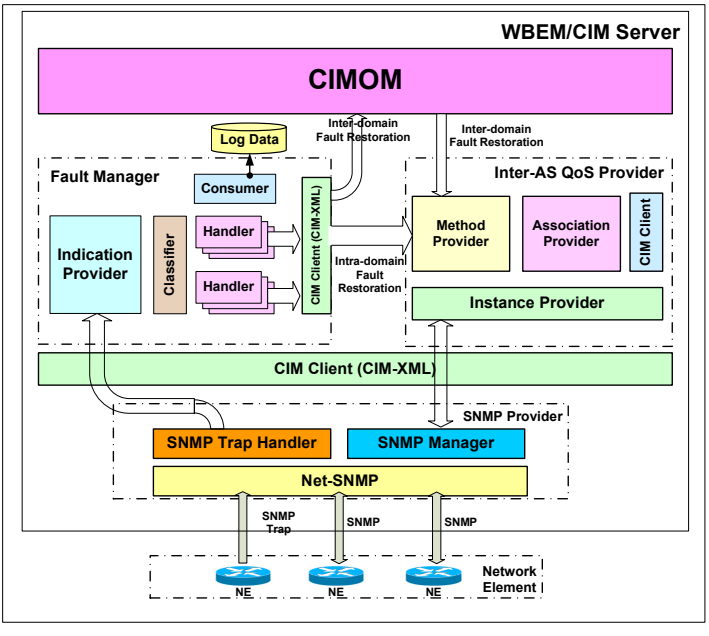


Fig. 3. Functional architecture of WBEM-based distributed fault management

WBEM Server has three components. SNMP provider, which includes SNMP trap handler to receive SNMP traps from SNMP agent supported network devices, and then sends that trap to fault manager. Fault Manager, upon receiving trap, classifies the fault, and logs to file or displays it. According to fault type it invokes local or ingress NMS's method provider, which includes fault restoration and fault correction functions. According to the configuration of fault management strategy, NMS redirects the user packet flow through the backup LSP.

The proposed WBEM-based distributed fault management system is basically designed for inter domain MPLS networks, but also includes intra domain fault management functions. Since we are considering inter-AS TE, distributed fault restoration schemes (i.e., path restoration) are used.

When there is any link/node/path failure, SNMP trap handler receives the trap from the network device. Using CIM-XML, it relays the trap to the Fault manager module, which in turn identifies trap, classifies the failure and generates the corresponding indication instance. Indication provider in the fault manager checks the content and creates necessary indication instance. Once the corresponding indication instance is created, it is subjected to process by the filters. The filters are classifying the faults, i.e. acting as a classifier. Based on the scope of the faults (fault classification and fault notification), the faults will be delivered to the corresponding handlers. Faults with respect to the intra-domain are locally handled and if fault requires end-to-end path restoration in inter-domain network, it should be delivered to the ingress NMS. Once fault is received by the CIM Server from remote fault manager, appropriate fault restoration method from inter-AS QoS provider is invoked.

3.2 CIM MOF-Based MO Design for Fault Management

Fig.4 shows the important CIM classes and associations designed for distributed fault management in inter-domain traffic engineering.

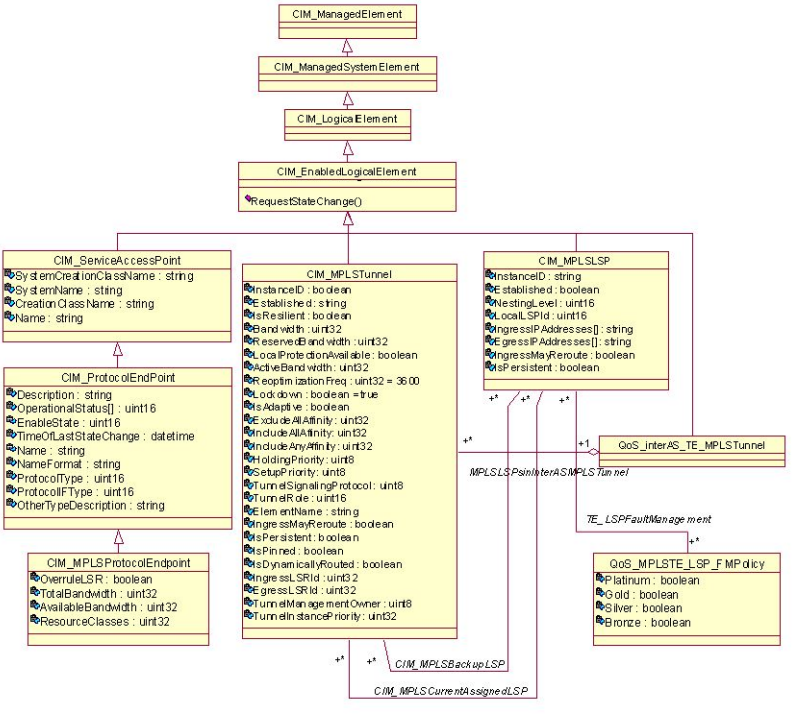


Fig. 4. DMTF CIM based MOF classes of MPLS LSP fault management

Classes with CIM prefix are supported by DMTF CIM, and CIM classes with QoS prefix are newly designed MOF-based extensions for fault management in inter domain networks. Most of the classes related to MPLS TE are derived from *CIM_EnabledLogicalElement* and they have *RequestStateChange()* method. For example, when *MPLSTunnel* down trap is received by the SNMP Trap handler, the association class *CIM_MPLSCurrenltyAssignedLSP* (shown in Fig.4) is enumerated to find the associated *CIM_MPLSLSPs* in that *CIM_MPLSTunnel* MO. The MO state is changed to reflect the current status update. Based on the trap location and severity, the fault manager notifies the trap to remote NMS InterASQoSProvider, where the *MPLSTunnel* is originated. When the recovery is done for MOs, the notification for recovery will also be sent to the concerned providers.

For each MPLS LSP, we can specify protection path option according to the protection requirement of differentiated user traffic, such as 1:1, 1:N, M:N. The backup LSP is defined by CIM_MPLSBackupLSP, which associates CIM_MPLSTunnel and CIM_MPLSLSP. During Service level agreement between the providers, the fault restoration method has also been negotiated and the backup path has been configured. The backup LSP from ingress NMS to egress NMS is usually specified to be SRLG (Shared Risk Link Group) – disjoint for the working LSP.

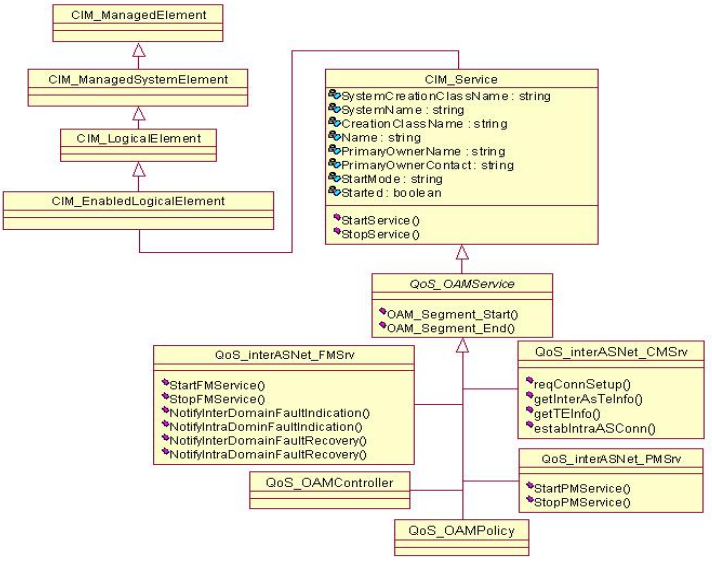


Fig. 5. DMTF CIM based MOF classes of MPLS OAM services

The fault management service is defined in *QoS_InterASNet_FMSrv* class (shown in Fig.5.), and it has the external methods for notifying the faults/recovery on managed objects. QoS*_OAMService* MOF has been designed for fault management. The *QoS_interASNet_PMSrv* is used to configure the start and stop of performance monitoring operation, and the collection of results statistics. The *QoS_interASNet_FMSrv* is used to configure the fault & alarm notification functions

for abnormal conditions on the established inter-AS TE-LSPs and also for fault restoration. The *QoS_interASNet_CMSrv* is used to AS connection establishment and gather traffic engineering information. Moreover *QoS_OAMPolicy* and *QoS_OAMController* classes were designed to define OAM policy for distributed fault management. All the classes are inherited from *QoS_OAMService* class.

The *CIM_IPProtocolEndpoint* MOs are associated with the link and the router's interfaces. For each router, which can be *QoS_ASBRouter* MOF (shown in Fig.6), it should have specific identification ID attached. This ID can be used as *ingressLSRID*/*egressLSR* ID in the MPLS Tunnels. *QoS_ASBRouter* contains *CIM_DeviceConnectivity_Collection* and *CIM_IPConnetivitySubnet* MOFs which are defined by the *CIM_HostedCollection*. These associations and collection classes represent the necessary MOs for link, node and MPLS TE_LSP, which are used in fault management.

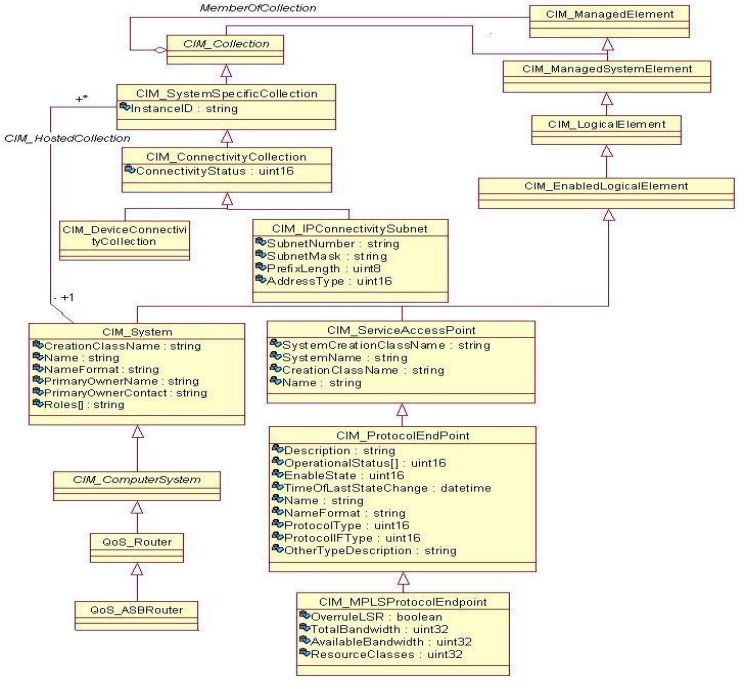


Fig. 6. DMTF CIM based MOF classes of physical node/link

3.3 Fault Restoration Procedure

Fault restoration procedure consists of three steps: fault detection step, fault notification step and fault restoration step. Fault detection is performed by physical layer detection capabilities such as Network Interface Card (NIC), MPLS signaling such as constraint based label distribution protocol (CR-LDP) and resource reservation protocol with traffic extensions (RSVP-TE) [8], and MPLS Operations, Administration, and Maintenance (OAM) functions [9].

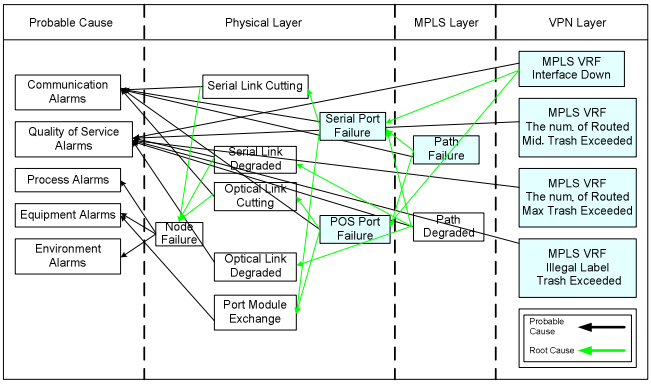


Fig. 7. Fault correlation

Fault notification message using the SNMP for accurate fault localization and diagnostics can be received from various network nodes such as router when NMS receives SNMP trap message. However, as SNMP notification is UDP based protocol, fault monitoring system may not receive all fault notification messages correctly, and may not be accurate.

Fig.7 shows root cause graph in fault correlation. In this figure, we classified fault occurrence's causes into three categories such as physical layer failure, MPLS layer failure, and virtual private network (VPN) layer failure. The shaded box in this figure is SNMP trap information and the others are known from each MOs. In fault localization module, basically we used trap messages for finding upper side of failure. For example, if path failure trap in MPLS layer is occurred, we continually analyze whether its MO of upper layer is failure and then we have confidence of root cause of failure.

Fig.8 shows the sequence diagram of an MPLS TE-LSP restoration. Physical layer fault recovery is also implemented in the same manner. When SNMP handler obtains SNMP trap from network device, it redirects this trap to fault manager's indication provider. Then indication provider creates instance of that trap and is redirected to

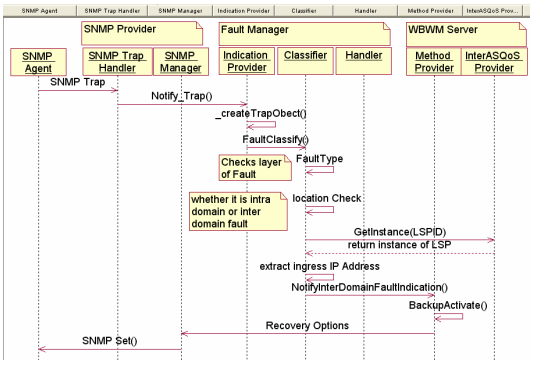


Fig. 8. Sequence diagram of MPLS TE LSP restoration

classifier function, which classifies fault type. According to fault type, it redirects to one of the registered handlers based on filters. Classifier checks two kind of parameters such as fault scope (intra domain or inter domain), and also type of fault (physical or MPLS layer). In the case of physical layer fault *QoS_ASBRouter* MO can be used to obtain working MPLS tunnel and accordingly failed LSP ID. In the second case, *CIM_MPLSLSP* MO can be used.

As mentioned earlier, handler invokes appropriate method of CIMOM method provider, which stores faults. Currently, four methods are used in fault restoration: *NotifyInterDomainFaultIndication(), NotifyIntraDomainFaultIndication(), NotifyInterDomainFaultRecovery(),* and *NotifyIntraDomainFaultRecovery().*

4 Implementation of Distributed Fault Management in WBEM-Based Inter-AS TE

We are currently implementing WBEM Server using OpenPegasus WBEM open source code in Linux platform, the MOF instances are implemented using C++. For the SNMP manager, we use Net-SNMP [6].

4.1 Fault Manager Implementation

The NMS of each domain is equipped with WBEM server and client functions for inter-AS traffic engineering. When fault manager module calls fault restoration function according to occurred fault (from local or remote NMS), it uses CIM-Client interface. The fault management operations of fault manager module for intra-AS traffic engineering are not open to other NMSs, but can be co-located with inter-AS traffic engineering modules.

In this implementation the MOFs are implemented in C++ objects, and are supported by instance providers. Currently we have implemented simple indication provider for fault manager module to handle and process the faults, and modified existing method provider by adding extrinsic functions, such as *NotifyInterDomainFaultIndication()* or *NotifyIntraDomainFaultRecovery()*, which are used for fault restoration.

4.2 SNMP Provider and SNMP Trap Handler Implementations

Since OpenPegasus does not support SNMP manager/agent model yet, SNMP Provider has been implemented based on Net-SNMP open source module, by integrating it to the WBEM architecture. When instance provider calls SNMP manager, SNMP provider acts as server, and using simple method provider interface it provides corresponding functionality to get information from SNMP Agent. While sending information/trap to inter-AS QoS provider or fault manager, it acts as client. When SNMP trap handler receives trap message from SNMP agent, it triggers fault manager by invoking *notifyTrap()* method of indication provider.

4.3 Performance Evaluations on Trap Handling

Currently we are implementing the proposed and designed distributed fault management for WBEM-based inter-AS TE. Initial analysis shows that, the trap

message will be delivered within 10-15 msec from SNMP trap handler to fault manager. Then the fault manager immediately generates indication and sends it to appropriate server by calling *NotifyInterDomainFaultIndication()* or *NotifyIntraDomainFaultIndication()* functions according to fault scope. With the help of filters and handlers, the local notification of the indication takes 60-80 msec, and in the case of remote CIM Server it takes 350-400 msec.

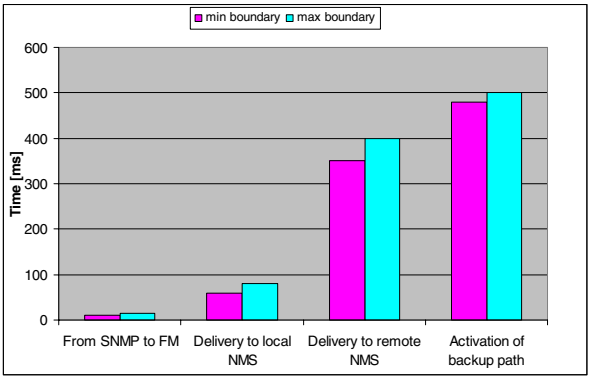


Fig. 9. Performance results

Notification about fault will be simultaneously delivered to the consumer also within 60-80 msec, and it prints out some log data (Fig. 9). Since the backup path is pre-configured, the end-to-end fault restoration does not take much time. The ingress NMS has backup pathinformation and it reconfigures the end-to-end path. As a result, we expected that the MPLS TE_LSP restoration will be completed within 1 sec.

5 Conclusion

In this paper we proposed design and implementation of distributed fault management for WBEM-based Inter-AS Traffic Engineering for QoS guaranteed DiffServ-over-MPLS using existing CIM MOFs with hierarchical inheritance. *QoS_OAMService* MOF, which designed for interASNET OAM functions, has been extended with fault management functionalities. We also extended existing providers with SNMP Provider, which is not supported by OpenPegasus and NET-SNMP has been integrated into WBEM Fault Management. Also the detailed interaction scenario among NMSs with WBEM server and client function of different AS domain networks for fault management have been designed. Currently we are implementing the proposed distributed fault management based OpenPegasus WBEM source code and Net-SNMP open source tool. From initial stage performance analysis, it takes less then 1 sec to detect, notify and restore the occurred fault.

Our future plan is to the integrate proposed distributed fault management with Service Level Agreement and improve the performance.

References

1. Open Pegasus, http://www.openpegasus.org/.
2. Osborne Simha, Traffic Engineering with MPLS, Cisco System, 2001.
3. Web-Based Enterprise Management (WBEM) Initiative http://www.dmtf.org/standards/wbem.
4. Douglas Mauro, Kevin Schmidt, Essential SNMP, O'Reily 2001.
5. Shanmugham S., A. Abdurakhmanov, Young-Tak Kim, **"**WBEM-based Inter-AS Traffic Engineering for QoS-guaranteed DiffServ Provisioning,**"** proceeding of IEEE BcN 2006, April 2006
6. Net-SNMP, http://net-snmp.sourceforge.net/.
7. Jae-Jun Jang, Young-Tak Kim, "Fault Management for GMPLS-based Optical Internet," Proceedings of Conference on APNOMS 2002, JeJu islands, Korea, September 2002.
8. Awduch et. Al, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF
9. Youngtak Kim, "MPLS OAM Functions and their Applications to Perform Monitoring, Fault Detection and Localization," proceeding of ACIS SERA03 (Internation Conference on Software Engineering Research, Management and Application), SanFrancisco, May 2003.
10. Sung-Jin Lim, Design and Implementation of Differentiated Backup Path Options for Fault Management in MPLS Networks, Master thesis, Yeungnam Univ., December 2003.

A Framework Supporting Quality of Service for SOA-Based Applications

Phung Huu Phu, Dae Seung Yoo, and Myeongjae Yi

School of Computer Engineering and Information Technology
University of Ulsan, Republic of Korea
{phungphu, ooseyds, ymj}@mail.ulsan.ac.kr

Abstract. Web Services and Service-Oriented Architecture (SOA) has been playing an important role as a middleware for interoperable transactions such as Business-to-Business and Enterprise Application Integration. Popular Web Services frameworks, such as Apache Axis, did not consider the Quality of Service (QoS), though these aspects are high demands in practice. In this paper, we present a framework supporting QoS built on the top of Axis. The framework is transparent with developers on both client and server side, and supports QoS including performance, accessibility, reliability, and security for SOA-based transactions. The design and implementation of our framework provide an easily integrated and flexibly extended approach to SOA-based applications. The key features and implementation methodology are described, with scenarios provided as usage examples of the framework.

1 Introduction

Service-Oriented Architecture (SOA) is the new trend in distributed systems aiming at building loosely-coupled systems that are extendible, flexible and fit well with existing legacy systems [1] [2]. SOA is based on Web Services framework, which intends to provide a standards-based realization of the service-oriented computing (SOC) paradigm. The real value of SOA is the ability to automate large-scale business processes, mixing a variety of technologies. SOA is based on Web Services technology and Web Services technology is based on XML; therefore, it is possible for any platform and programming language to build applications using Web Services. These features of Web Services technology can solve the weaknesses of other distributed technologies such as language and platform dependence, inflexibility, disruption to existing interfaces of old systems [3].

In the SOA-based approach, each system specifies the interoperability with others for the integration and machine-to-machine interactions. The transactions of interoperability are created as services using Web Services/SOAP interface [6]. Services can be published for integration communications thanks to UDDI specification. Each system, depending on particular business rules, might communicate with other systems by requesting services via SOAP/HTTP transactions.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 232–241, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In general, SOA approach gains the flexibility and scalability, extensibility as well as trust and security in interoperability.

Although SOA supplies an architecture for software interoperability in effective manner, there are many challenges that need to be investigated to develop the infrastructure for such an architecture. One of the challenges is Quality of Service (QoS) for SOA-based transactions. Normally, developers use a Web Services framework to build SOA-based applications since such frameworks supply simple approach for creating, deploying as well as consuming Web Services. However, most Web Services frameworks have not been considered the aspects of QoS; therefore, it is very difficult for developers when considering these aspects in their applications. In this work, two aspects of QoS in SOA-based transactions are considered: reliability and security.

In our point of view, to support QoS in SOA-based applications, a framework for service management is needed. A common framework identifies specific functions that need to be addressed in order to achieve decentralized interoperability. The aim of this research is to build a framework that supports QoS for SOA-based applications. The proposed framework supports both client side and server side and is designed in extensible and flexible manner based on open standards supporting Web Services transactions. Developers could use the framework by transparently configuring framework mechanisms and could manage services on concrete scenarios.

The rest of this paper is organized as follows. We present the design of modules in the framework in section 2. In section 3, the implementation of the framework is detailed. Section 4 gives the evaluation and shows how the framework can be used by giving some real scenarios using our framework. We conclude the paper and specify future work in section 5.

2 The Design of the Framework

2.1 Goals

Recently, there are a number of Web Services frameworks but they were not considered QoS. For instance, Apache Axis framework [4] only provides the way for constructing SOAP processors such as clients, servers, gateways. GLUE framework offers a more simple invocation model by generating an interface automatically and providing it remotely, i.e. only supports client side [11]. Asynchronous Invocation Framework [15], which is built on top of Axis and provides asynchronous invocation on client side without using asynchronous message protocols, or Web Services Invocation Framework (WSIF) [12] support only on client side and did not consider QoS for transactions. Therefore, the goal of this work is to propose a framework that supports QoS properties for SOA-based transactions. The main aim of our proposed framework is that the transactions of services are guaranteed the reliability and security. The aspects are transparent with application layer; thus, Web Services applications built on this framework are guaranteed the reliability and security without considering these aspects in application code. In reliability aspect, the framework deals with the problems such

as system or network failures in transactions of the architecture and resolves the problems by managing and recovering failed long-running transactions. In the view of security, data transferred within the framework are guaranteed the integrity, confidential, and non-repudiation. Access control of services is also considered via authentication mechanism. Besides, the proposed framework also aims to resolve the following issues such as: providing the better performance of Web Services applications, giving simple and flexible usage on client side and server side, and supporting for multiple Web Services implementations.

Moreover, a framework design should support the heterogeneity of services, infrastructures, and protocols, as this is the main design goal of Web Services. And as the technology continues to evolve, a number of specifications are being proposed to address the areas necessary to support Web Services. Therefore, the design of framework should provide the extensibility, which is important feature of a framework for extending additional functions in future.

2.2 Modules

Figure 1 shows the structure of our proposed model for the framework. There are three main blocks in this framework: interoperability control, security control and reliability control. The purpose of interoperability control is to create SOAP message based on service description and business rules. The SOAP message is transferred to security control module if a secure exchange of message is required; otherwise, the message will be transferred to reliability control module. Since the interoperability control is supported by most popular Web Services frameworks such as Axis or WSIF, our design does not investigate on this module. The following subsections present the detailed design of security and reliability modules in our framework.

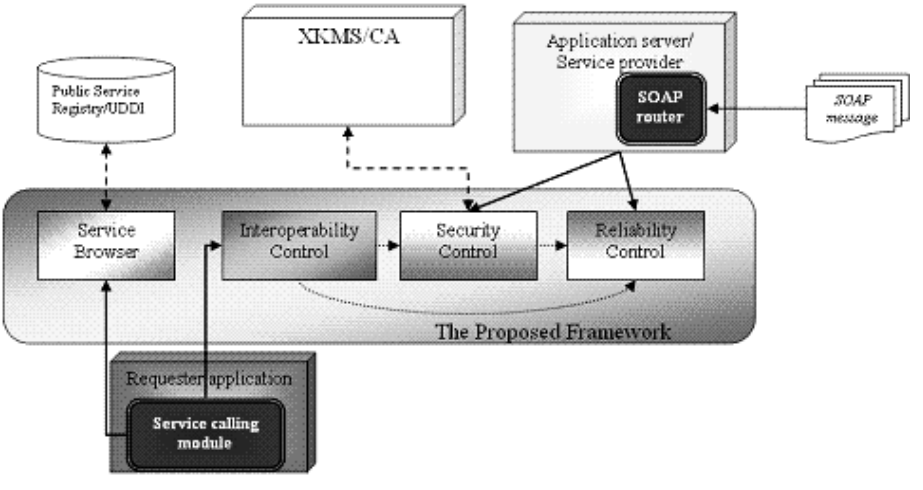


Fig. 1. The structure of the framework

Security Module. A Web application needs to be defended with a number of threats. In this work, the framework is built for Web Services transactions which use SOAP message protocol working over the HTTP protocol. Supposing that HTTP relies on a secure network (i.e. the network is guaranteed the security in the perimeter, network, host layer), this work focuses on security challenges of application layer in interoperable transactions in SOA infrastructure. In this case, SOAP messages need to be protected to ensure a high degree of confidential and integrity.

In this framework, a security model that combines of XKMS [7], XML Encryption [8] and XML signature [9] has been proposed for transactions in SOA. This model follows WS-Security [10] by using credentials in SOAP message with XML Encryption and XML Signature. In this model, XKMS is used as a protocol for public key distribution center. The XKMS host plays the role as a CA in interoperable transactions to distribute and to check the trust of public key in XML signature and as a key distribution center for public key cryptography using XML Encryption.

Since using XKMS, which is similar to Public Key Infrastructure, each participant in this model must have key pair (public and private key) and register the public key to a XKMS server. Thus, data transferred in this model inherits its authentication, data integrity and non-repudiation of PKI architecture and XML signature. It also supplies confidential thanks to XML Encryption. XML information of transactions stored in each system is used to check after transactions, warrantee auditing of security policy. This model fulfills the requirement of Web Services interoperability in SOA architecture.

Reliability Module. Reliability is one of the important QoS properties for Web Services transactions. This module deals with the problems of failed Web Services requests over time, duplicated request, and the ordered delivery of messages. Since Web Services transactions are transferred normally on HTTP, which provides only best-effort delivery, it is necessary to consider reliability in SOAP message transactions. The structure of SOAP message offers the way to fulfill reliability by describing additional information and inserting it to SOAP message headers. Although WS-Reliability [5] (former version is WS-ReliableMessaging) is the specification proposed to guarantee the reliability of SOAP message delivery, it is much sophisticated for deployment. More, these specifications did not provide solutions to incorporate with other properties of QoS such as security, accessibility and performance. In this framework, reliability module is designed on the basic of the Messaging Model [5].

Based on this model, additional fields are added to SOAP message headers in our framework. On client side, these fields are embedded into the request message before sending to Web Services provider:

MessageID: used to identify the request message to avoid the duplication request.

Sequence Number: used in the case of re-sending an invocation which has been failed before

Time-to-live: time within which the client program should receive an acknowledgment or result from the server provider from invoking a request.

This module also provides asynchronous invocation for client application for better performance and loose coupling, i.e. the client should not depend on the processing times of the Web Services. To provide these properties, this module will be implemented in multi-thread model for communicating reliably with server. When timeout event happen without receiving expected response, this module re-invokes the request until the result is received.

On server side, upon receiving the request message, the module extracts Reliability fields (*MessageID* and *Sequence Number*) to check for duplication. The fault message is sent back to client if the request is duplicated, otherwise, the module dispatches SOAP message to an appropriate handler for invoking the requested service, starts its timer. The module sends back data to appropriate module when receiving result from requested service or timeout event from timer. Fields for managing reliability are also added to headers of the response message before sending back to another module for returning result to the client.

Additional Properties. In the context of Web Services, in addition to security and reliability which are solved by proposed modules, performance, accessibility, availability and integrity are other properties of Web Services needed to be investigated. *Integrity* property is gained in security module where data transferred is guaranteed the confidential, integrity and non-repudiation. Our proposed framework supports monitoring mechanism, therefore, it allows for a weak control of *availability* by checking times of unavailability and long response in monitoring log. *Performance* is an important QoS property in Web Services. Performance is influenced by client and server processing times and network latency. In this framework, more processing time is required in modules, therefore, we apply the multi-thread and thread pooling paradigm to reduce overhead in transactions.

3 The Implementation of the Framework

3.1 Implementation Methodology

To build a framework for Web Services from scratch is a huge work and is not the main aim of this research. Our implementation focuses on the QoS as mentioned, therefore, the implementation should be built on top of a general framework. For that purpose, we choose the popular Apache Axis framework for extending additional functions to support QoS properties as designed.

Apache Axis is a distributed computing framework, much like CORBA, DCOM, RMI, for Web Services. Axis shields Web Services developers from the details of dealing with SOAP and WSDL. Developers could use Axis on the server side to write service and deploy (using tools in the framework) it as a web application on a web server; also, developers could use Axis on the client side to consume Web services by using simple APIs. Based on the architecture

of Axis, our proposed framework is implemented by extending the Axis both on client and server side to gain the intended properties of QoS. Since Axis is built on Java, it could be used in multiple platforms thanks to this property of Java language.

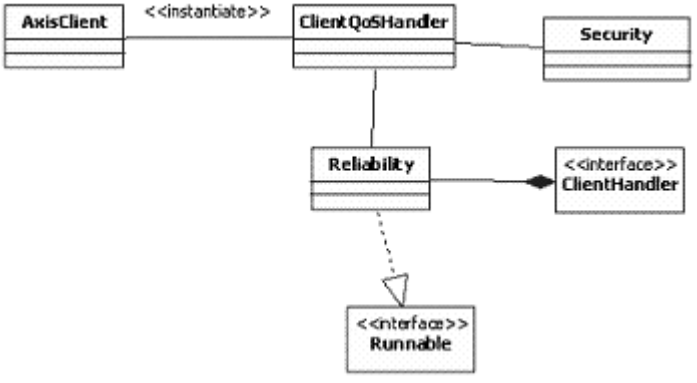


Fig. 2. The framework on client side

3.2 The Implementation of the Framework on Client Side

In Axis framework on client side, data after processed into SOAP messages is dispatched to class AxisClient to invoke a service request via Handlers in Axis. In our framework, some classes are added to the Axis framework on client side in order to control the quality of service for Web Services transactions. The figure 2 shows the relationship of the implementation.

The aim of class ClientQoSHandler is to handle the quality of services for the proposed framework on client side. When receiving message from AxisClient class, this class instantiates the class Security, which handles security module, and class Reliability, which handles reliability module and performance for client side of the framework. The ClientQoSHandler invokes security module to encrypt and sign the SOAP message for the invocation. The algorithm for encrypting and signing is loaded from *a configuration file* which is exchanged by partners in advanced. The message returned from security model is dispatched to Reliability class. This class implements the Runnable interface and associates handler object to process the result back to the client thread.

3.3 The Implementation of the Framework on Server Side

On server side, our framework also is implemented on the top of popular Axis framework. In this implementation, MessageContext received from AxisServer is dispatched to ServerQoSHandler which is responsible for handling the aspects of quality of service on server side within the framework. The figure 3 shows the relationship of modules in the customized framework.

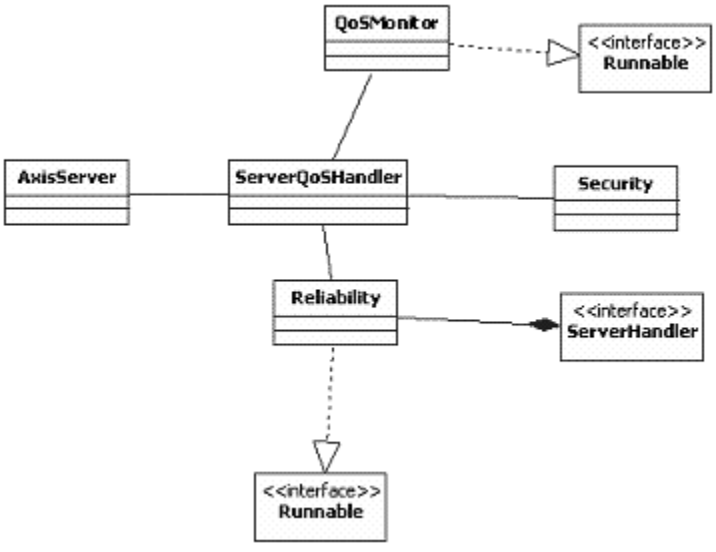


Fig. 3. The framework on server side

The functionalities of additional modules on server side are similar to those on client side. Security module handles MessageContext dispatched from ServerQoSHandler to de-encapsulate (validates the signature and decrypts SOAP message according to designed security model). Reliability module interacts with Handlers on server side. This module implements Runnable interface and be responsible for performance considerations as showed in the following section. Besides, the framework on server side implements a QoSMonitor module to monitor Web Services execution and performance.

3.4 Performance Considerations

To reduce the overhead in the framework, multi-thread and thread pooling mechanism is used in implementation both on client and server side.

On client side, multi-thread technique provides an asynchronous invocation so that the framework can improve performance since the client could resume its work after sending an invocation.

On server side, on receiving a request, the request message is dispatched to the QoS Handler before invoking the appropriate remote object; new thread is instantiated to process QoS properties as designed. The multi-thread mechanism can be used in order to process concurrent invocations, however, it also incur more overhead due to instantiating the threads. Thread pooling technique [15] can be used to reduce this overhead. In this technique, threads are shared in a pool, when needing the handlers can get a thread-worker from the pool and then release a thread back into the pool when finishing. The pool eagerly acquires a

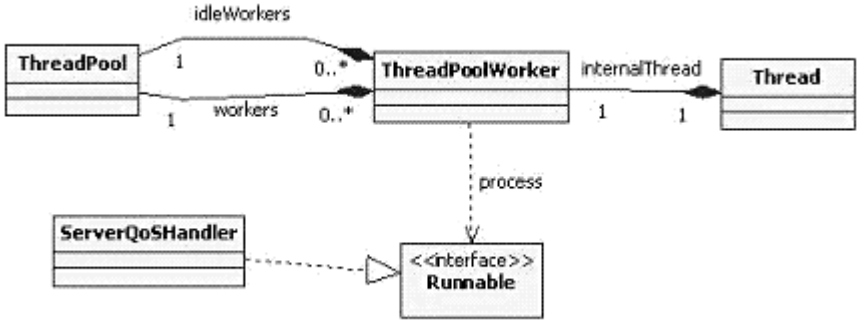


Fig. 4. Thread pooling used in the framework

pre-defined number of workers. If the demand exceeds the available resources, the pool would instantiate more resources to execute. Figure 4 shows thread pooling model used in the framework.

4 The Evaluation of the Framework and Case Studies

In the context of security, the framework provides the authentication, integrity and non-repudiation for transactions thanks to that data is signed using XML Signature and keys are certified by XKMS system. The confidential of data is also ensured by XML Encryption. In this framework, various cryptography algorithms can be used by editing configuration files, without rebuilding the framework.

In the context of reliability, the framework provides fault-tolerant for transactions in SOA-based applications. The framework on client side could monitor requests, and guarantee that any requests receives appropriate response message, otherwise, it tries to re-send requests or report the fault for system recovery. Fields added into reliability headers in SOAP headers make sure that messages are transacted in a reliable environment. Reliability module also provides asynchronous invocation for client application for better performance and loose coupling, i.e. the client should not depend on the processing times of the Web Services.

Since our framework is built on top of Axis framework, we can use all functions of Axis. Axis framework is implemented in Java language; therefore, it can run in any application server supporting Java. Developers could use APIs of the framework on both client and server side in the similar way to Axis (see [4] for detail).

To demonstrate the use of our framework in practice, we have developed a number of scenarios. For example, a two-tier transactions application in Internet banking transactions has been developed using our framework. Once using the framework for Web Services applications, transactions transferred among the Internet banking systems are guaranteed QoS properties provided by the

framework. A long-running transactions application scenario has also considered and built based on our framework. In this scenario, security is guaranteed in message-level and can be configured in a file without customizing the framework. The properties of security such as integrity, confidential and non-repudiation are guaranteed in transitive manner through point-to-point transactions between each two contiguous systems. Reliability of transactions is also guaranteed in point-to-point manner in a long-running transaction. The framework provides asynchronous invocation on client side; therefore the performance penalty of such a long-running transaction could be reduced. Clients can significantly resume faster its work when transaction is dispatched to a next system. Multi-thread and thread pooling mechanism on server side of the framework offers the better performance for Web Services transaction processing. Readers can refer [14] for more details.

5 Conclusion and Future Work

This work proposes a framework supporting QoS for Web Services transactions in SOA infrastructure. The transactions in SOA-based applications built on top of this framework are guaranteed the security and reliability without caring of application developers. In addition, the framework on client side is designed and implemented in multi-thread mechanism so that a client can significantly resume faster its work, therefore, the performance penalty of Web Service can be reduced, especially in long-transaction Web Services applications. On server side, the properties of availability and performance are also reached by using thread pooling technique. The implementation of the framework is based on top of popular Axis framework, thus, the framework automatically inherits Axis's heterogeneity regarding communication protocols and back-ends of Web Services.

Providing end-to-end QoS in a heterogeneous, potentially cross-organizational SOA is a particular challenge and still needs to be investigated. In the future, we will investigate on QoS for SOA-based applications both on qualitative and quantitative. We will refine and evaluate QoS properties in particular application areas so that these aspects can be applied in real world SOA-based applications. QoS negotiation and agreement, contracts, composition of QoS requirements and appropriate evaluations will be also considered in future work.

Acknowledgements

This research was partly supported by Research Fund of University of Ulsan and the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government. The authors also would like to thank Ulsan Metropolitan City and the Network-based Automation Research Center (NARC) which partly supported this research. The authors also thank the anonymous reviewers for their carefully reading and commenting this paper.

References

1. Hao He, "What Is Service-Oriented Architecture," 2003, http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html
2. M.P. Papazoglou and D. Georgakopoulos, "Service-oriented computing," *Communications of the ACM, Vol.46, No. 10*, 2003, pp. 25-28.
3. David Booth et al., "Web Services Architecture," W3C Working Group Note 11, February 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/
4. Apache Software Foudation, "Apache Axis," http://ws.apache.org/axis/, 2006.
5. Web Services Reliability (WS-Reliability), 2004. http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/
6. SOAP Version 1.2 Part 1: Messaging Framework, http://www.w3.org/TR/soap12-part1/, W3C Recommendation (2003).
7. XML Key Management Specification, http://www.w3.org/TR/xkms/
8. XML Encryption Syntax and Processing, http://www.w3.org/TR/xmlenc-core/
9. XML Signature, http://www.w3.org/Signature/
10. SOAP Security Extensions: Digital Signature, http://www.w3.org/TR/SOAP-dsig/
11. Markus Volter, Michael Kircher, Uwe Zdun, "Remoting Patterns," *John Wiley & Sons*, 2005.
12. Apache Software Foudation, "Web Services Invocation Framework," http://ws.apache.org/wsif/, 2006.
13. Phung Huu Phu, Myeongjae Yi, "A Service Management Framework for SOA-based Interoperability Transactions," *In Proceedings of the 9th Korea-Russia Intl. symposium on Science and Technology (KORUS2005, IEEE Press)*, Novosibirsk, Russia, 2005, pp. 680–684.
14. Phung Huu Phu, "Quality of Service for Interoperable Transactions in Service-Oriented Architecture," *Master Thesis*, University of Ulsan, South Korea, May 2006.
15. Uwe Zdun, Markus Vlter, Michael Kircher, "Design and Implementation of an Asynchronous Invocation Framework for Web Services," *In Proceedings of the CWS-Europe* (2003), pp. 64-78.
16. H., Santhosh K., Jen-Yao C., "A Service Management Framework for Service-Oriented Enterprises," *In Proceedings of the IEEE International Conference on E-commerce Technology*, California, July 06-09, 2004. pp. 181-186.
17. Y. Huang and J. Chung, "A Web Services-based Framework for Business Integration Solutions," *Electronic Commerce Research and Applications 2 (2003)*, pp.15-26.
18. G. Wang at el., "Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architectures," *In Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conf (EDOC 2004)*, California, September 20-24, 2004. pp. 21-32.

Performance Improvement Methods for NETCONF-Based Configuration Management*

Sun-Mi Yoo[1], Hong Taek Ju[2], and James W. Hong[3]

[1] Samsung Electronics, Korea
sunmi.yoo@samsung.com
[2] Dept. of Computer Engineering, Keimyung University, Korea
juht@kmu.ac.kr
[3] Dept. of Computer Science and Engineering, POSTECH, Korea
jwkhong@postech.ac.kr

Abstract. IETF's NETCONF WG has taken efforts in standardizing configuration management protocol, which allows high interoperability of configuration management. In addition to interoperability, high performance is also required in configuration management, but many researches have often discarded the performance aspect of it. In order to fill that void, this paper proposes methods to improve performance with layers of NETCONF. It demonstrates and compares the performance evaluations to verify the efficiency of the proposed methods. This evaluation can be used as a guideline to effectively implement NETCONF. Moreover, it also demonstrates the process of performance evaluation of configuration management.

1 Introduction

The network configuration management sets up operation values of devices that constitute the network and collects and analyzes the values. For example, it sets up routing parameters of routers or security values of firewalls and monitors the values. A centralized server manages various remote devices for configuration management, which is essential on current networks. IETF has proposed the Network Configuration (NETCONF) [1] standard for configuration management of remote devices. The NETCONF standard [2] assumes that the current network is composed of various devices from diverse vendors. These standards can formulate the remote configuration management more effectively.

Along with interoperability, the efficiency of configuration management is an important factor to consider. Since the configuration management is carried out against many devices, the efficiency is required thus more. For instance, when monitoring the values of many devices, an efficient process is mandatory to achieve

* This research was supported in part by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2005-C1090-0501-0018) and by the Electrical and Computer Engineering Division at POSTECH under the BK21 program of the Ministry of Education, Korea.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 242 – 252, 2006.
© Springer-Verlag Berlin Heidelberg 2006

correct values. Moreover, fast configuration management is essential to avoid causing distress in networks when modifying operation values of devices. The NETCONF standard has been explored for functional requirements in various ways, whereas there is little discussion on the efficiency issue. So far, the performance enhancement methods for configuration management using NETCONF have yet to be proposed in any literature. Furthermore, NETCONF performance evaluation result and its analysis, along with the methods for efficient use of NETCONF have not been discussed, while existing technical reports on NETCONF implementation do not report on its performance.

This paper proposes several methods that improve NETCONF protocol performance. NETCONF can be conceptually partitioned into four layers by its functions. This paper considers three core layers that have a great effect on NETCONF performance; the application protocol layer, the RPC layer and the operations layer. These three layers are used to propose methods to improve performance in each layer and the performance is evaluated to verify the effect of the methods. The proposed methods include the ones that follow the original NETCONF standards, as well as the ones that do not. In addition, since the method of approaching by layers is applied, these methods can be adopted to various configuration environments.

To achieve statistical significance, we have measured the response time, network usages, and computer resource usages of each layer. Our methods manage smallest network usages to reduce the response time and the load on network, which produces rapid configuration management. Furthermore, computing resource usages are reduced to enable the manager system to manage many devices, and the NETCONF agent system to apply to the embedded devices.

The remainder of this paper is organized as follows. Section 2 introduces NETCONF implementations and research of network management performance. Section 3 presents the testing environment and architecture of XCMS used for the test. Section 4 presents the performance evaluation results of transport protocol layer. Section 5 presents the performance evaluation results of RPC protocol layer. Section 6 presents the performance evaluation results of operations layer. Finally, we summarize our work and discuss future work in section 7.

2 Related Work

In this section, we present implementations of XML-based network configuration management using NETCONF by various earlier works. We also present related work on network management performance.

2.1 Implementations of NETCONF

The NETCONF standard is an early stage that does not have many implementations yet. This section briefly introduces a system that is implemented by our previous work and an open source project called EnSuite [7].

In our previous work, we have developed an XML-based configuration management system, XCMS [8], which implemented the first internet draft of IETF NETCONF for IP network devices. XCMS used the XPath [8], which has been standardized since the fourth draft and SOAP over HTTP [3] as a transport mechanism. XCMS supports the latest internet draft of IETF NETCONF protocol [2] as well as three transport protocols; SOAP over HTTP [3], BEEP [5] and SSH [4]. In XCMS, a centralized manager controls the configuration information of network devices equipped with NETCONF agents. XCMS can manage configuration information of diverse devices depending on the proprietary operating systems of the vendors.

EnSuite [7] is a network management platform prototype based on NETCONF and is an open source project from LORIA-INRIA in France. LORIA-INRIA had developed a NETCONF agent called YENCA, in their previous work. YENCA was implemented by C language but had limited functions. LORIA-INRIA then developed a new configuration management system, EnSuite, which has improved upon its functions and architecture. EnSuite consists of a NETCONF web-based manager, a NETCONF agent and a set of extension modules that are implemented in Python.

2.2 Research on Network Management Performance

This section discusses research work that focus on the performance of network management using XML technologies.

Aiko Pras et al have presented Web Services for management on the performance differences between SNMP and Web Services-based management [13]. To compare their performances, they investigated bandwidth usage, CPU time, memory requirements, and round trip delay. To conduct the tests, they implemented several Web Services-based prototypes and compared their performance to various SNMP agents. These tests showed that there is a significant difference in the bandwidth requirements of SNMP and Web services. They concluded that SNMP is more efficient for cases where only a single object is retrieved although Web Services-based management is more efficient for larger number of objects. Thus, Web Services-based management is more suitable for large scale networks.

Another interesting study on the performance of network management has been conducted by Apostolos E. Nikolaidis et al [12]. This study mentions that the Universal Plug and Play (UPnP) protocol undertakes the Lan configuration and management. Due to relatively high bandwidth and the limited number of devices in this paper, the traffic issues are of secondary importance. They examined the unnecessary management traffic volume that may occur due to the verbose nature of XML technology, used by the protocol. Their solution exploits some capabilities provided by the protocol and uses real-time compression in order to effectively reduce the management traffic, while keeping the response times at the same or even lower levels. The solution mainly comes from the application of the Lempel-Ziv compression algorithm, with minimal additions in the proposed DSL Forum standard. They evaluate the performance and usability of the solution, implementing a typical configuration example with the CPE WAN management protocol [12].

3 Measurement Environment and Implementations

We applied our XML-based configuration manager and agent, XCMS, as NETCONF implementations. Our XCMS manager and agent are connected by a 100Mbps Ethernet and run on Linux servers with Pentium IV 2.4GHz CPU and 512MB RAM. The XCMS agent manages network information of Linux system configuration; 'interface'. For the statistical significance, we have measured the response time and memory usage of XCMS manager and agent for 1000 times and averaged the results.

The NETCONF standards specify a communication process of the XCMS manager and the agent as follows: First, the manager attempts to correspond with the agent. When the session is opened, each peer (both the manager and the agent) must send a <hello> element containing a list of the peer's capabilities. If the <hello> element does not have any wrong elements, the session is a NETCONF session and a manager begins sending NETCONF request messages. Each transport protocol has a different process for opening a NETCONF session. Hence, we only measured the communication between NETCONF request messages and response messages.

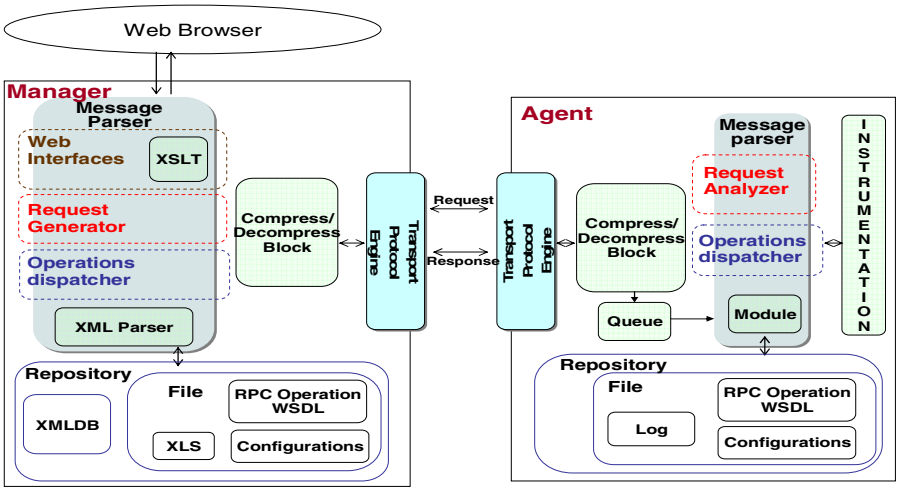


Fig. 1. Architecture of XCMS

As mentioned earlier, the experiments are performed by using our XML-based configuration management system; XCMS. Figure 1 illustrates the architecture of XCMS consisting of a manager and an agent. The XCMS manager and the agent are implemented in C. They use ZLIB for compression and decompression and open source implementations for transport protocol. They use gSOAP implemented with C/C++ languages to exchange SOAP over HTTP RPC messages and Roadrunner implemented with C/C++ languages for BEEP and a port forwarding method for SSH. We used the libxml, which is the lightweight one among existing XML parsers as an XML parser in the manager and the agent. The XCMS agent is implemented with the latest NETCONF drafts and we have added our proposed solutions to XCMS.

4 Transport Protocol Layer

The transport protocol layer of NETCONF provides a communication path between the manager and the agent. The NETCONF protocol allows multiple sessions and transport protocols. The NETCONF protocol [2] is currently considering three separate transport protocol bindings for transport; Secure Shell (SSH) [4], Block Extensible Exchange Protocol (BEEP) [5] and SOAP over HTTP [3]. Both the functions and the performance of the protocols are examined to select a transport protocol for configuration management. This section demonstrates how the performance of transport protocol affects the overall performance, and proposes a mechanism to improve the application protocol layer. Finally, the performance is evaluated to verify the effect of the proposed mechanism. We have performed several 'get-config' operations, which read configuration information of various sizes to check the transport protocol performance. The manager constructs a request message, which performs 'get-config' operation with a subtree filtering and sends the message to the agent. We operated the network interface configuration management for this experiment. The number of network interface is increased in intervals of 2 due to the changed information on sizes.

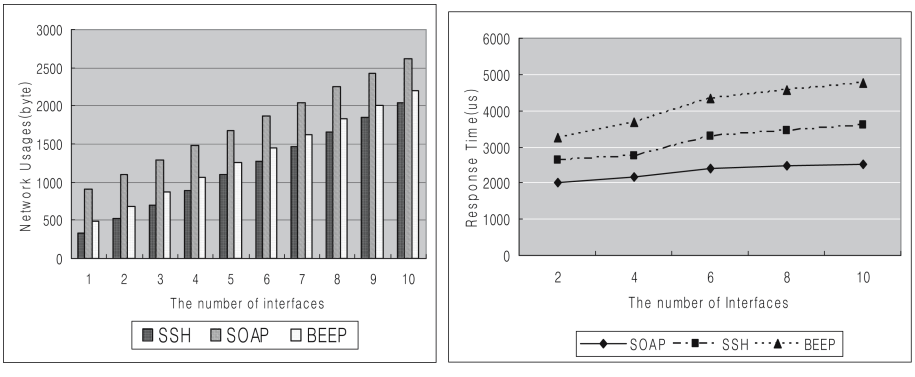


Fig. 2. Network usages and Response time by Transport Protocols

The network usages and the response time have been measured to examine NETCONF transport protocol performance. First, each transport protocol's network usage is examined. SOAP over HTTP has the heaviest network usages, whereas SSH has the least, as illustrated in Figure 2 left graph. Both SOAP over HTTP and BEEP are based on XML technology, and they both append extra messages for header contents. Despite the fact that a message conveys only one parameter and one value, the size of the corresponding XML message is bigger.

Next, by using each transport protocol, the response time of configuration management is measured. In our analysis, the response time is defined as the time interval between sending and receiving a NETCONF message. Although it was predicted that the result of the response time would be similar to the network usage, it has been found to differ, as Figure 2 right graph demonstrates.

The implementation methods are considered as the reason for such result. The XCMS manager and agent use the port forwarding mechanism for SSH. It is a simple implementation mechanism but due to the port forwarding process, some overhead occurs. However, the actual time difference is very little in the consideration of the unit of time and the time is round trip time.

Although the response time of the three transport protocols are similar, the network usages are quite different. When the network usages in configuration management have an immense effect on the network traffic, a mechanism for reducing the sizes of NETCONF messages is needed. For solution, we used the compression method since the repetitive nature of text patterns in typical XML message are produced by NETCONF. Moreover, many papers have also proposed the compression mechanism for reducing the sizes of XML. We have compressed payloads using the ZLIB library.

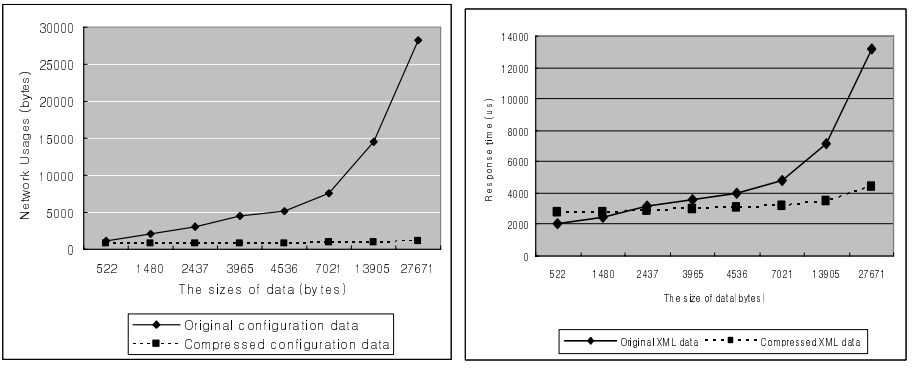


Fig. 3. The Network Usage and Response time of NETCONF Messages Compression

Experiments were conducted with the increasing sizes of messages to test the effect of the compression mechanism. The response time and the network usage were measured. Figure 3 left graph illustrates that the size difference between the original data and the compressed data increases as the size of the data increases. Figure 3 right graph shows that the response time depends on the network usage. Although the process for compressing data produces overhead time, the large size data can ignore this fact, as the number of packet fragmentations reduces.

5 RPC Layer

The NETCONF uses an RPC-based communication model. The RPC layer provides a simple, transport-independent framing mechanism for encoding RPCs. The <rpc> element is used to enclose a NETCONF request sent from the manager to the agent. Next, the <rpc-reply> element encloses a NETCONF message sent in response to the <rpc> operation on the RPC layer. The <rpc> element can only have a NETCONF method and should have one-to-one communication with the response message. The NETCONF provides two mechanisms for operating commands, with no pipelining and pipelining. With no pipelining, a NETCONF manager waits for a response

message of the previous request before sending next request message. This mechanism has some inter-request time and low efficiency. Therefore, NETCONF provides pipelining in order to lower the elapsed time and to improve the efficiency. The pipelining mechanism serially sends request messages before the previous requests have been completed, instead of waiting on a response with pipelining. Furthermore, the NETCONF agent must send the responses only in the order the requests were received. We investigate the effect of pipelining and propose a mechanism for improving the efficiency on RPC layer.

We increased the number of requests containing a NETCONF command to measure the effect of the pipelining mechanism. The command processes the 'get-config' operation to obtain interface information and has a response message of around 465bytes. The response time is measured by SSH and the other protocols have the same result. Figure 4 demonstrates that the performance of pipelining is better than the one of no pipelining. However, the pipelining mechanism has some risks. The configuration management process could be destroyed if a request message is corrupted during processing.

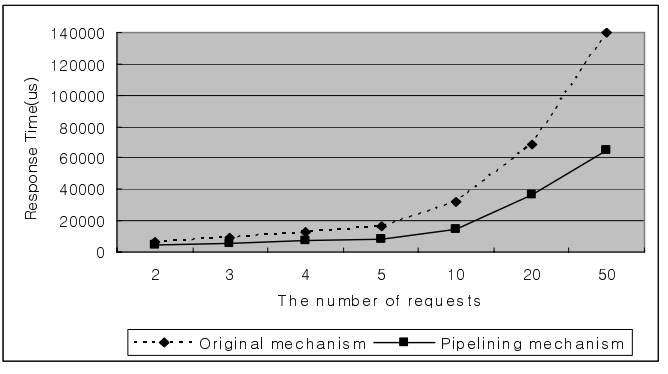


Fig. 4. No Pipelining/Pipelining response times (μs)

The pipelining mechanism does not improve the network usage performance. The number of packets at a point of time on the network increases compared to the case without the pipelining mechanism. The NETCONF protocol draft states that the <rpc> element only has the NETCONF operation for its sub element. However, several NETCONF operations related with each other are needed for a completed configuration management. For instance, in order to complete setting the configuration, operations for obtaining the current data as well as for setting the new data are required.

We propose the Multi-Command mechanism for improving the efficiency of network usage. Multi-Command is a NETCONF request message with several commands under an <rpc> element. For example, the following three operations can be put into a <rpc> element with Multi Command: an <edit-config> operation to <candidate> data, <commit> operation for applying the <candidate> data to the <running> data and <get-config> operation for checking the result of previous operations. An agent sequentially provides the Multi-Command mechanism processes

on the request operations and creates a response. This mechanism also has some risks that are similar to the pipelining mechanism. If any errors occur while processing a request message, all operations of the request message are canceled by the <roll-back> function.

The network usages and the response time have been measured to verify the effect of the Multi-Command mechanism. We have performed a similar process to the experiment of the pipelining mechanism and compared to three communication mechanisms on a RPC layer. Three protocols show similar results of using the Multi-Command mechanism, which does not have a great effect on SSH, since it adds a small header to a payload. However, SOAP over HTTP and BEEP show larger difference of network usages. Moreover, the communication process of the multi command mechanism and the pipelining mechanism is similar. This mechanism also has no inter request time but has a little additional processing time.

We have compared our results with the pipelining of HTTP. The NETCONF pipelining has similar effect to the HTTP pipelining on response time. The requests are buffered before transmission so that multiple HTTP requests can be sent with the same TCP segment. As a result, HTTP pipelining does not generate unnecessary headers and reduces the number of packets required to transmit the payload. On the other hand, the NETCONF pipelining uses the RPC communication and transfers a request message containing an operation at once. The NETCONF pipelining cannot reduce the number of packets or network usages. The proposed Multi Command mechanism uses the RPC communication and transfers a request message containing several operations at once. Therefore, this mechanism can reduce the number of packets and the total network usages similar to the HTTP pipelining.

6 Operation Layer

The NETCONF operations for managing configuration information of devices are processed on an operation layer. The NETCONF protocol provides filtering methods in selecting particular XML nodes. In particular, both of <get> and <get-config> operations use two different filtering mechanisms. One is the subtree filtering, stated as default by NETCONF specific and another is the XPath capability of NETCONF protocol. These two mechanisms have their own advantages and disadvantages. The subtree filtering is easy to comprehend but difficult to implement correctly, since the NETCONF draft has ambiguous pats and is short of examples of accuracy on its usage.. On the other hand, the XPath allows common XPath expressions. Thus, it is simple to implement, but rather difficult to comprehend.

Subtree filtering and the XPath demonstrate the difference of performance. NETCONF WG has pointed out that the XPath has a heavy memory usage and response time for configuration management, and suggested subtree filtering as a substitute for the XPath. The XPath requires loading the whole XML message to DOM document before applying the request. This process efficiently creates XML messages, but the XPath uses many memory usages regardless of complexity and sizes. NETCONF WG has proposed that the subtree filtering is lighter than the XPath and has the same results with the XPath. We have experimented in order to verify these claims to confirm the difference of performance between the subtree

filtering and the XPath. We have referenced other researches [15] of the difference between the subtree filtering and the XPath and compared with ours. We used the <get-config> operation and two sets of equivalent requests. The 'Request Message_1' shown in Figure 5 needs to merge different processes. The result of this request includes all interfaces whose name equals 'eth0' along with the interface names which 'mtu' equals '1000'. The subtree filtering and the XPath expression are 'Request Message_1' in Figure 5. The 'Request Message_2' shown in Figure 5 needs to parse simply and does not need to merge. The result of this request includes the names of all interfaces.

Figure 5 demonstrates our experiment results of processing time and memory usauge. The two experiments have produced rather different results. In the first experiment, the subtree filter takes a longer time than the XPath, with the merging. However, in the second experiment, the XPath takes a longer time than the subtree filtering, without the merging. Moreover, these two experiments have different XPath results. As mentioned above, the XPath [9] builds a DOM document and then applies the XPath expression on the DOM tree. It can travel efficiently on the DOM tree for selecting property nodes and shows more power in complicated structure. In contrast, the subtree filtering is reliant on our implementation, which uses a recursive top down approach for selecting nodes. The subtree filtering process is repeated when different nodes need to be merged, which takes more time. The XPath uses the same memory usage on both of two experiments, but it uses more memory than the subtree filtering.

	Subtree Filtering	XPath
Request Message_1 (Using merge)	<filter type="subtree'> <interfaces> <iface> <name>eth0</name> </iface> <iface> <mtu>1000</mtu> <name/> </iface> </interfaces> </filter>	<filter type='xpath'> //name[.='eth0']|//mtu[.='1000'] </filter>
Processing Time	2.085 ms	1.901 ms
Memory Usage	1816 KB	1828 KB
Request Message_2 (Using simple)	<filter type="subtree'> <interfaces> <iface> <name/> </iface> </interfaces> </filter>	<filter type='xpath'> //name </filter>
Processing Time	1.716 ms	1.954 ms
Memory Usage	1812 KB	1828 KB

Fig. 5. Subtree Filtering/XPath

7 Concluding Remarks

In this paper, we have proposed mechanisms for effective reduction of traffic volume, response time and computer resource usage in configuration management using NETCONF. We have also provided an analysis of performance results by NETCONF layers in our implementation; XCMS. In particular, we have investigated network usages, memory requirements and response times.

The response time of transport protocol layer is hardly affected by the transport protocols. The network usage of transport protocol layer is the sum of the request/response message that is used to manage configuration information and the headers of each transport protocol. Also, according to our experiments, clearly, the response time depends on the network usages. We presented the compression mechanism for reducing both the network usages and the response time. We also compared our experiments to other researches of XML compression.

RPC layer provides the pipelining mechanism for more efficient configuration management. The pipelining mechanism can reduce the total response time, but it cannot affect the network usage. We proposed the solution of Multi-Command for reducing total network usage, which is similar to HTTP pipelining. Our solution reduces the response time as well as the network usages from our measurements. Naturally, this mechanism follows a RPC communication method.

We have measured and compared the processing time and the memory usages of the XPath and subtree filtering in our implementation. The XPATH is suitable for processing messages since it requires fewer sources in merging the results as well as in embedding the systems.

References

1. IETF, “Network Configuration,” http://www.ietf.org/html.charters/netconf-charter.html.
2. R. Enns, “NETCONF Configuration Protocol”, draft-ietf-netconf-prot-11, http://www.ietf.org/internet-drafts/draft-ietf-netconf-prot-11.txt, February 2006.
3. T. Goddard, “Using the Network Configuration Protocol (NETCONF) Over the Simple Object Access Protocol (SOAP),” draft-ietf-netconf- soap-06, http://www.ietf.org/internet-drafts/draft-ietf-netconf-soap-06.tx t, September 16, 2005.
4. M. Wasserman, T. Goddard, “Using the NETCONF Configuration Protocol over Secure Shell (SSH)”, http://www.ietf.org/ internet-drafts/draft-ietf-netconf-ssh-05.txt, Oct. 2005.
5. E. Lear, K. Crozier, “Using the NETCONF Protocol over Blocks Extensible Exchange Protocol,” http://www.ietf.org/internet-drafts/draft-ietf-netconf-beep-07.txt, Sept. 2005.
6. R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, L. Masinter, P. Leach and T. Berners-Lee, “Hypertext Transfer Protocol - HTTP/1.1”, RFC 2616, IETF HTTP WG, June 1999.
7. INRIA-LORIA, EnSuite, http://libresource.inria.fr/projects/ensuite.
8. Hyoun-Mi Choi, Mi-Jung Choi, James W. Hong, “Design and Implementation of XML-based Configuration Management System for Distributed Systems,” Proc. of the IEEE/IFIP NOMS 2004, Seoul, Korea, April 2004, pp. 831-844.
9. W3C, “XML Path Language (XPath) Version 2.0,” W3C Working Draft, November 2005.
10. W3C, "Web Services Description Language (WSDL) Version 1.2" July 2002.

11. Sun-Mi Yoo, Hong-Taek Ju, James Won-Ki Hong, “Web Services Based Configuration Management for IP Network Devices,” Proc. of the IEEE/IFIP MMNS 2005, LNCS 3754, Barcelona, Spain, Oct., 2005, pp. 254-265.
12. Apostolos E. Nikolaidis et al, “Management Traffic in Emerging Remote Configuration Mechanisms for Residential Gateways and Home Devices,” IEEE Communications. Magazine, Volume 43, Issue 5, May 2005, pp. 154-162.
13. A. Pras, T. Drevers, R. v.d. Meent and D. Quartel, “Comparing the Performance of SNMP and Web Services-Based Management,” IEEE eTNSM, Vol. 1, No. 2, Dec. 2004, pp. 1-11.
14. Mi-Jung Choi et al, “XML-based Configuration Management for IP Network Devices,” IEEE Communications Magazine, Vol. 41, No. 7, July 2004. pp. 84-91.
15. Vincent Cridlig, et al, "A NetConf Network Management Suite:ENSUITE", Proc. of the IEEE IPOM 2005, LNCS 3751, Barcelona, Spain, Oct., 2005, pp. 152-161.
16. Henrik Frystyk et al, "Network Performance Effects of HTTP/1.1, CSS1, and PNG," W3C, June 1997.

Zone-Based Clustering for Intrusion Detection Architecture in Ad-Hoc Networks

Il-Yong Kim, Yoo-Sung Kim, and Ki-Chang Kim

School of Information & Communication Engineering, Inha University
253 Yonghyundong Namgu Incheon, Korea
bush@super.inha.ac.kr, {yskim, kchang}@inha.ac.kr

Abstract. Setting up an IDS architecture on ad-hoc network is hard because it is not easy to find suitable locations to setup IDS's. One way is to divide the network into a set of clusters and put IDS on each cluster head. However traditional clustering techniques for ad-hoc network have been developed for routing purpose, and they tend to produce duplicate nodes or fragmented clusters as a result of utilizing maximum connectivity for routing. Most of recent clustering algorithm for IDS are also based on them and show similar problems. In this paper, we suggest to divide the network first into zones which are supersets of clusters and to control the clustering process globally within each zone to produce more efficient clusters in terms of connectivity and load balance. The algorithm is explained in detail and shows about 32% less load concentration in cluster heads than traditional techniques.

Keywords: Intrusion detection architecture, Ad-hoc Network, Clustering.

1 Introduction

Detecting intrusion in ad-hoc networks is harder than in regular networks. In wired or LAN/WAN, we have a gateway where network traffic is concentrated, and an IDS(Intrusion Detection System) can be installed there. In ad-hoc networks, we don't have such a convenient point. A candidate for IDS installation in ad-hoc networks would be a node that has relatively large number of neighboring nodes located within communication range. There should be multiple of them since we have to cover all participating nodes in the network, and these nodes need to communicate with each other to convey the local intrusion-related information. Finding such nodes and allowing them to exchange information efficiently is not easy. To make the situation worse, the nodes are mobile, and the network topology can change frequently: we may have to repeat the hard process of setting up intrusion detection architecture again and again.

The problem of computing efficient clusters and maintaining them in spite of frequent changes in network topology is studied in relation with routing in ad-hoc networks. For routing purpose, the most important parameter that determines the efficiency of a cluster is connectivity, and the suggested techniques tend to allow

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 253 – 262, 2006.
© Springer-Verlag Berlin Heidelberg 2006

duplicate nodes (nodes belonging to more than one cluster at the same time) and to produce many fragmented clusters (clusters with only one or two members). For intrusion detection point of view, the critical parameter should be the number of nodes since the cluster head is responsible for collecting security-related data from all member nodes. Many researchers studied the above clustering problem for Ad-Hoc IDS[1-4]. However the existing clustering techniques for IDS are simple adaptations of ones used in Ad-Hoc routing and still have the similar problem of duplicate nodes and fragmented cluster.

We propose a zone-based clustering technique for Ad-Hoc IDS. The aim is to avoid duplicate nodes or fragmented clusters and to control the size of clusters to prevent excessive load in cluster heads. Our technique clusters the given network of nodes in two steps. In the first step, the network is divided into a set of sub-networks, called zones. Clusters are formed within a zone in the second step; that is no cluster is formed across zones. Zoning helps us to group geographically adjacent nodes; clustering within this zone is a much easier problem than clustering for the whole networks. Zoning also helps us to maintain the clusters. The replacement of a cluster head can be handled within the corresponding zone only. The proposed algorithm has been implemented and tested using GloMoSim simulator[5], and the result shows a significant reduction in the cluster header's packet processing load.

The rest of the paper is organized as follows: Section 2 examines previous studied on Ad-Hoc IDS clustering, Section 3 explains the proposed clustering techniques in detail, Section 4 describes experimental results, and Section 5 contains the concluding remarks.

2 Related Work

[6] classifies IDS architectures for Ad-Hoc networks into Stand-Alone, Distributed and Cooperative, and Hierarchical. In Stand-Alone type, all nodes have IDS, but each node performs intrusion detection independently. A watchdog program can be combined with Stand-Alone architecture [7]. It can monitor the packet exchange between adjacent nodes and prevent communication with suspicious nodes. Distributed and Cooperative type also allows all nodes to have IDS, but they should cooperate in collecting global intrusion-related data [1].

Hierarchical architecture is one that uses clustering technique [2, 3, 4]. Fig. 1 shows an example of Hierarchical architecture. Each cluster has a cluster head, and adjacent clusters share a gateway node as shown in Fig. 1(a). IDS's are installed in cluster heads as shown in Fig. 1(b). Various techniques have been developed to build an efficient cluster and to select a cluster head. [3] suggests a technique in which the number of neighboring nodes is computed for all nodes and one with the maximum number becomes a cluster head. Neighboring nodes are computed based on pre-determined number of hops: if the number of hops is 2, all nodes within 2 hops from a node become neighboring nodes for that node. Once a cluster head is determined, the head itself and the surrounding nodes (located within the pre-determined number of hops from the head) form a cluster. The same process to find a cluster head and the corresponding cluster, then, repeats for the rest of nodes. [4] suggests to build cliques first. A clique is a set of nodes that are within 1 hop from each other and corresponds

to a cluster in general term. A random cluster head, then, is elected from the clique. To protect the identity of a cluster head (for security reason), all nodes within a clique have the same probability to be elected.

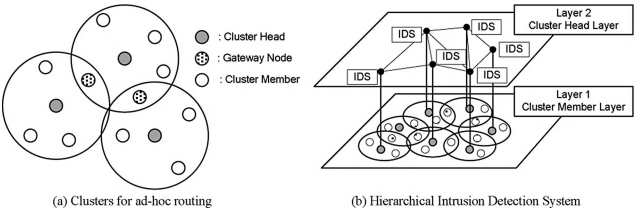


Fig. 1. Clustering for IDS

Clustering for the purpose of intrusion detection is reported to be effective in reducing the load on CPU. To achieve the same detection rate, clustering technique helps to reduce the CPU usage by 29% compared to the case when no clustering is performed [4]. However, we still observe unnecessary monitoring overhead in previous techniques such as duplicate nodes and excessive number of small cluster fragments. Duplicate nodes happen in the gateway nodes existing in the overlapped area between two adjacent clusters, and they increase the overhead on cluster heads because of multiple monitoring by at least two neighboring cluster heads. Fragmented clusters are generated since the clustering algorithm prefers a set of well-connected nodes as a cluster and tends to leave boundary nodes not belonging to any cluster as single-node clusters.

3 Zone-Based Clustering Technique

We prevent cluster fragmentation by first dividing the network of nodes into zones. A zone is a set of nodes that are located close to each other. Clusters are built within each zone not allowing inter-zone clusters. Node sharing between clusters is prohibited to prevent duplicate nodes.

3.1 Zoning

Zoning process is distributed: seed nodes that are located at the boundary of the network announce themselves as initiators and start to build zones at the same time. We assume all nodes in the network know their neighbor nodes within one hop before starting the zoning process. A node becomes a seed node if it has 1 or 2 neighbors; we prefer lonely nodes as starting points to avoid the formation of fragmented clusters. The seed nodes start to form zones following the algorithm shown in Fig. 2.

In the beginning all nodes are themselves zones, a one-member zone. A node can be in several states; initially all nodes are in SLEEP state as shown in Fig. 3. A zone must have a coordinator which determines whether to merge or not with neighbor zones. A coordinator that actively starts the zone-merging process is called an initiator; one that passively responds to this merging request is called a participant.

So, zoning process is a negotiation process between two coordinators that represent two neighboring zones: one of them is an initiator, the other a participant. The state changes of these two nodes are shown in Fig. 3.

1. Send *setup_init* to all neighbor nodes belonging to different zones other than the current one.

2. Collect *setup_request* from them for a pre-determined time. Each *setup_request* message has the zone size the sender belongs to. This and the arrival time of message will be used to compute its priority. The messages are inserted to a message queue in the priority order.

3. Pop a message from the queue, and mark all the nodes in the zone the sender belongs to as a new member of the current zone. Repeat this process until the pre-determined zone size is reached. If the maximum zone size has been reached, exit the zoning algorithm, and start the clustering algorithm.

4. Send *coordinator_election_request* to all members in the current zone.

5. Collect *coordinator_election_reply* from them for some pre-determined time. Each *coordinator_election_reply* message includes a number for neighbor nodes that are not in the current zone.

6. Elect one with the maximum such number. Report this node as a new coordinator to all members.

Fig. 2. Zoning algorithm performed by the coordinator

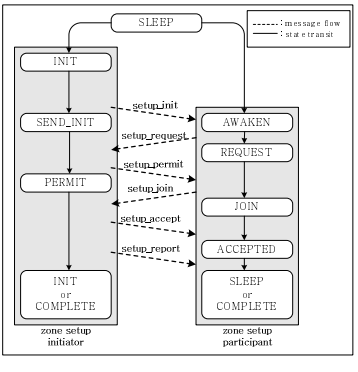


Fig. 3. State transition of a node

When zoning process begins, the seed nodes send *setup_init* messages to their neighbors. The seed nodes in this case act as initiators, and the neighbors become participant. The participants send back the size of the zone they belong to. The initiator examines the size and determines whether to merge or not. The basic criterion is zone size after merging; if it becomes greater than some threshold, the merging is abandoned.

Merging decision may be made for several neighbor zones at the same time. The *setup_init* messages are sent to all neighboring coordinators, and if multiple responses arrive within the same time period the initiator would consider all the corresponding

participants as possible candidates for merging. The response from the participant is a *setup_request* message, and it contains the responder's zone size. The initiator use this information to give a preference to smaller participants: it is again to deter the formation of fragmented clusters. The participant does a similar selection based on the size of a zone. The *setup_init* message also contains the zone size of the initiator, and when the participant has received a number of *setup_init* messages, it prefers the initiator with the smallest zone size and sends *setup_request* to it. The merging process is shown in Fig. 4(a), and the preference of the smaller zone is shown in Fig. 4(b).

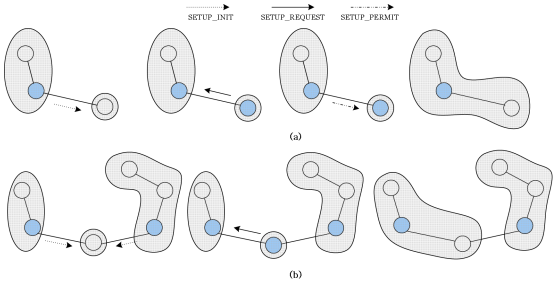


Fig. 4. Zone building

When a new zone is formed through merging, a new coordinator has to be elected. For this purpose, the old coordinator broadcasts a *coordinator_election_request* message to all zone members. Each zone member responds with the neighbor node table it has. The old coordinator counts only the neighbor nodes that do not belonging to the current zone, and the member with the largest neighbor node number (excluding ones within the current zone) will be elected as the new coordinator. The node number of the new coordinator will be notified to all zone members, and this new coordinator will start the next phase of zone merging process.

Merging process stops when all the neighbor zones are merged to the initiator's zone or when the size of zone has reached the maximum. When the maximum size is reached, we have a final zone, and the state of all nodes within this zone becomes COMPLETE. When the maximum size is not yet reached, we repeat the process of zone merging again.

3.2 Zone Building Example

Fig. 5 shows an example of zone building process. Initially there are 26 nodes in 900m × 400m area. All of them are coordinators and form a zone by themselves as explained in Section 3.1. When the zoning process begins, node 0, 3, 10, and 24 become the initiators because they have less than or equal to 2 neighboring nodes as shown in Fig. 5(a). These nodes send *setup_init* messages to their neighbors (or participants) and merge them. The initiators are shown in shaded circles. They are also shown in the tables of nodes in the right side of the figure. For each table, the shaded entry is the initiator and other entries are the participants for the initiator. The result is shown in Fig. 5(b). In the figure, 4 new zones and the corresponding new coordinators of them (in shaded circle) are shown.

The new coordinators tend to exist at the boundary of the zone since they have a more number of neighbor nodes (neighbors belonging to the same zone are excluded when computing the number of neighbor nodes). These new coordinators act as initiators to start the next merging process. This process of merging and selecting a new coordinator is repeated again and again until the size of all zones reach the maximum or there is no more neighbor zone to merge. The final zones are shown in Fig. 5(f).

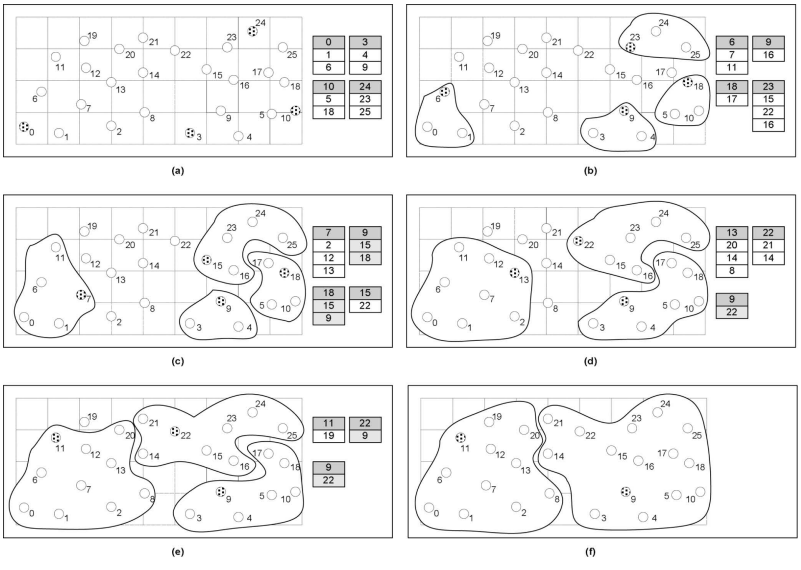


Fig. 5. An example of zone building process

3.2 Clustering

Clustering is the process of selecting a cluster head and including the neighbor nodes of this head into the cluster. For security reason, an outsider should have no clue about who will be and is the cluster head, and, therefore, all nodes should have an equal chance of becoming a cluster head. Once the head is elected, the neighbor nodes are included to the corresponding cluster until the maximum size is reached. Sharing of nodes between clusters is prohibited, and collecting nodes from other zone is also prohibited.

Clustering is performed at each zone independently. The coordinator at each zone at the time of completion of the zone will act as the zone manager. This manager performs the algorithm in Fig. 6.

Basically it repeats the process of selecting a cluster head and building a cluster. In the beginning, it sends *cluster_init* message to all zone members. All members respond with a random number as shown in Fig. 7. The node which sent the biggest random number will be chosen as a cluster head. In Fig. 7, there are 5 nodes in the zone, and node 4 has the largest random number. Since it has neighbor node 3 and 5 as shown in figure, the first cluster will be node 3, 4, and 5 with node 4 being the head. The clustering manager repeats the same process for the rest of nodes. The

second table in Fig. 7 shows node 1 and 2 become the second cluster with node 1 being cluster head.

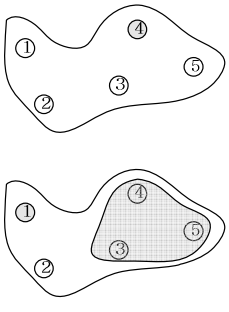
1. Send *cluster_init* to all zone members.

2. Collect *cluster_init_reply* from them for some pre-determined time. Each *cluster_init_reply* message contains the number of neighbor nodes and a random number the sender has generated. Put all sender nodes in a table with the number of neighbor nodes and the random numbers.

3. In the table, mark a node with the maximum random number as a cluster head. Mark the neighbor nodes of this cluster head one by one as cluster members until there are no more neighbors or the cluster maximum size is reached.

4. Remove the cluster head and its cluster members from the table, and go to step 3. If the table becomes empty, exit the algorithm.

Fig. 6. An algorithm performed by the zone manager



random	Node ID	Neighbor
363	4	3, 5
327	3	4, 5, 2
258	1	2, 3
150	5	4, 3
112	2	1, 3

random	Node ID	Neighbor
258	1	2
112	2	1

Fig. 7. Clustering process

3.3 Zone/Cluster Maintenance

The zones and clusters are dynamic entities. Their members may move out of them, or new members come and join. More seriously, the zone manager or the cluster head may move out of the region, or we may need to replace them for security reasons[1]. All these events may require rebuilding of the zone or the cluster. Rebuilding is processed in the form of merge or split – a zone or cluster whose size is below some minimum threshold will be merged to the neighbor zone or cluster while one whose size becomes greater than some maximum threshold will be split. The moving-out or moving-in of a node is detected by short-term or long-term hello message. A short-term message is a one-hop broadcast. All nodes issue short-term messages regularly to detect the membership change. Upon receiving, it all nodes respond with their zone and cluster identifiers. A long-term message is issued by the zone manager to all cluster heads in the corresponding zone periodically. The cluster heads responds with a message containing updating information such as membership change.

[1] In fact, we replace the zone manager and the cluster head periodically to avoid their exposure to persistent packet observers.

The merging of a cluster is initiated by the cluster head whose cluster size has shrunk below some threshold. It sends a merge request to its neighbor cluster head. When two clusters are merged successfully, a report about this change will be sent to the zone manager by the combined cluster head. The split of a cluster is also initiated by the corresponding cluster head. The cluster head sends a split request to the zone manager, and the manager will start the cluster-forming algorithm in Fig. 6, but in this case only for the members in the corresponding cluster. The merge or split of a zone is initiated by the corresponding zone manager. Merging is essentially the same process as the zone building process in Fig. 2.

The moving of a cluster head or a zone manager is dealt with a new election. A new cluster head is elected when the members do not hear a short-term message from the head for some time period. They broadcast a random number to each other, and one issued the largest number will become the new head. Similar election is performed at zone level when the current zone manager disappears; again this absence of a manager is detected by the silence of the long-term message.

4 Experiments

We have implemented our zone-based clustering technique in simulated network of mobile nodes using GloMoSim [5]. On the same network we also have implemented two previous clustering techniques for IDS in mobile network proposed in [4] and [8]. The aims of experiments are two folds. First we compare the cluster size, and secondly we compare the load in cluster heads. Cluster size is an important parameter to evaluate the performance of IDS. To avoid traffic concentration on a few cluster heads, the size should be evenly distributed among the cluster heads. Also to reduce inter-cluster-head traffic, the number of clusters should be controlled, and for this reason fragmented clusters (that has only one or two members) should be avoided as much as possible. Finally to avoid unnecessary traffic monitoring, node sharing between clusters should be prevented.

The result for cluster size comparison is shown in Table 1. In the table, WCL and CIDS represent the clustering technique in [8] and [4] respectively, and ZIDS our technique. Three figures are compared: number of clusters, average cluster size, and number of single cluster. For each category, we varied the number of participating nodes by 30, 50, and 100. The first column in the table shows the number of clusters produced by each technique. "n" shows the number of nodes in the network. As can be seen in the table, ZIDS is producing the least number of clusters. The next column shows the average cluster size. This data becomes meaningful when combined with that in the third column, which shows the number of fragmented clusters produced by each technique. For example CIDS is producing lots of fragmented clusters, but the average size is between those of WCL and ZIDS. This means that CIDS is producing two kinds of clusters most of time – very large ones and very small ones. Very large clusters will penalize the cluster head; very small ones will increase traffic between cluster heads. On the other hand WCL produces almost no fragmented clusters as shown in the third column. However its average cluster size is relatively high.

Table 1. Result for cluster size comparison

	Number of Clusters			Avg. Cluster Size			Number of Single Clusters		
	n=30	n=50	n=100	n=30	n=50	n=100	n=30	n=50	n=100
WCL	15	18	35	2.71	4.40	7.90	1	0	0
CIDS	14	17	33	2.45	4.00	4.90	3	6	16
ZIDS	13	15	32	2.89	2.71	3.32	4	3	7

Fig. 8 shows the numbers of packets monitored by cluster heads in CIDS and ZIDS. The traffic was generated using CBR(Constant Bit Rate) application. We defined a pattern file that contain 50 CBR traffic pattern, generated 10 such files, and applied them to 50 and 100 node network respectively. As can be seen in figure, ZIDS shows the minimum packet monitoring load: the amount of packets in ZIDS is about 32% less than that in CIDS.

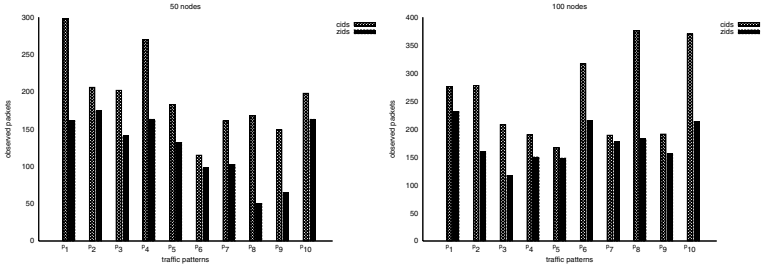


Fig. 8. Number of monitored packets for 50 and 100 node network

5 Conclusion

In this paper, we have proposed a zone based clustering technique for intrusion detection in Ad-Hoc network. Clustering process is essentially a distributed process since it is hard to control all the nodes in a mobile network. However, by dividing the network into a set of zones that contain geographically close nodes, we can control the clustering process globally within each zone and produce more efficient clusters. This zoning helps to produce clusters with evenly distributed size; it also facilitates better management of clusters when the nodes move across the cluster boundary. We have measured the performance of our technique in terms of traffic load on cluster heads which was about 32% lighter than that in traditional clustering techniques.

References

1. Y. Zhang and W. Lee: Intrusion Detection in Wireless Ad-Hoc Networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networks (MobiCom), Boston, USA (2000)
2. D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C-Y. Tseng, T. Bowen, K. Levitt and J. Rowe: A General Cooperative Intrusion Detection Architecture for MANETs. In: Proceedings of the third IEEE International Workshop on Information Assurance (IWIA), College Park, MD, USA (2005)

3. O. Kachirski and R. Guha: Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks. In: Proceedings of the 36th Hawaii International Conference on System Science (HICSS), Hawaii (2003)
4. Y. Huang and W. Lee: A Cooperative Intrusion Detection System for Ad Hoc Networks. In: Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), Fairfax, VA, USA (2003)
5. GloMoSim Simulator's web site: http://pcl.cs.ucla.edu/projects/glomosim/
6. P. Brutch and C. Ko: Challenges in Intrusion Detection for Wireless Ad-hoc Networks. In: 2003 Symposium on Applications and the Internet Workshops (SAINT), Orlando, Florida, USA (2003)
7. S. Marti, T. Giuli, K. Lai, and M. Barker: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: Proceedings of 6th International Conference on Mobile Computing and Networking (MobiCom), Boston, USA (2000)
8. M. Chatterjee, S. K. Das, and D. Turgut: WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Journal of Cluster Computing, 5(2), (2002)
9. L. Zhou and L. J. Hass: Securing ad hoc networks. In: IEEE Networks, 13(6), (1999)
10. Li. Y and J. Wei: Guidelines on Selecting Intrusion Detection Methods in MANET. In: The Proceedings of ISECON 2004, Newport, (2004)
11. M. Bechler, H.-J. Hof, D. Kraft, F. Pählke, and L. Wolf: A Cluster-Based Security Architecture of Ad Hoc Networks. In: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Hong Kong, China (2004)
12. C.-K. Toh: Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall PTR, (2002)

Tracing the True Source of an IPv6 Datagram Using Policy Based Management System*

Syed Obaid Amin[1], Choong Seon Hong[2,**], and Ki Young Kim[3]

[1,2] School of Electronics and Information, Kyung Hee University,
1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701 Korea
obaid@networking.khu.ac.kr, cshong@khu.ac.kr
[3] Electornics and Telecommunications Research Institute,
161 Gajeong-dong,Yusung-gu, Daejeon, 350-700, Korea
kykim@etri.re.kr

Abstract. In any (D)DoS attack, invaders may use incorrect or spoofed IP addresses in the attacking packets and thus disguise the factual origin of the attacks. Due to the stateless nature of the internet, it is an intricate problem to determine the source of these spoofed IP packets. This is where; we need the IP traceback mechanism i.e. identifying the true source of an IP datagram in internet. While many IP traceback techniques have been proposed, but most of the previous studies focus and offer solutions for DDoS attacks done on IPv4 environment. Significant differences exist between the IPv4 and IPv6 Networks for instance, absence of option in basic IPv6 header. Thus, the mechanisms of IP Traceback for IPv4 networks may not be applied to IPv6 networks. In this paper, we extended our previous work i.e. PPM for IPv6 and removed its drawback by using Policy Based IP Traceback (PBIT) mechanism. We also discussed problems related to previously proposed IPv4 traceback schemes and practical subtleties in implementing traceback techniques for IPv6 networks.

Keywords: DDoS, Traceback, IPv6, Network Security.

1 Introduction

To deter (D)DoS attacks, technologies like Intrusion Detection System (IDS) [13], Intrusion Prevention System (IPS) [14] and the Firewalls [15] are good solutions. However, in reality, prevention of all attacks on the internet is nearly impossible and the situation gets worse due to anonymous nature of IP protocol i.e. an attacker may hide its identity if he wants to. Moreover, the routing decisions are taken on destination addresses and none of the network unit makes sure the legitimacy of source address. Therefore, when prevention fails, a mechanism to identify the source(s) of the attack is needed to at least ensure accountability for these attacks and here we need the traceback techniques.

The elements that were threatening for IPv4 networks can also be intimidating for the future IPv6 network. To cope with IPv6 networks, we need to modify IPv4's

* This work was supported by MIC and ITRC Project.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 263 – 272, 2006.
© Springer-Verlag Berlin Heidelberg 2006

traceback technologies to be suited to IPv6 network. The reasons behind this amendment are the technological differences between these two network-layer protocols for instance, change in header size or fragmentation mechanism.

As mentioned before, the goal of traceback scheme is to identify the true source of a datagram. To achieve this task we try to pass the info of a packet or a path taken by a packet to the victim. One of the ways is that routers probabilistically or deterministically mark path information in packets as they travel through the Internet. Victims reconstruct attack paths from path information embedded in received packets. Packet marking techniques can be subdivided in Deterministic Packet Marking (DPM) [10] and Probabilistic Packet Marking (PPM) [2, 3, 4, 9]. In messaging routers probabilistically send ICMP messages, which contain the information of forwarding nodes the packet travels through, to the destination node. Victims reconstruct attack paths from received ICMP messages [1]. Another way of tracking the source of a packet is Packet Digesting in which routers probabilistically or deterministically store audit logs of forwarded packets to support tracing attack flows. Victims consult upstream routers to reconstruct attack paths [5, 8].

In this paper, we start our discussion with our previous work i.e. PPM algorithm for IPv6 networks [17]. Later on, to eliminate the deficiency of IPv6 PPM, we propose an IP traceback mechanism using Policy Based Management System. The rest of this paper is articulated as follows: In section 2, we describe related work. Section 3 outlines our previously proposed technique [17]. Section 4 covers the IP traceback technique using Policy Based Management System. Section 5 provides the simulation results and finally, we summarize our findings in Section 6.

2 Related Work

2.1 Packet Marking

Packet Marking [1][3][4][10] algorithms are based on the idea that intermediate routers mark packets that pass through them with their addresses or a part of their addresses. Packets can be marked randomly with any given probability or deterministically. The victim can reconstruct the full path with given mark packets, even though the IP address of the attacker is spoofed. This scheme was improved in several different ways; some of them introduced improved coding methods and security. All of the IPv4 marking algorithms suffered by the space limitation of IPv4 header. Therefore they have to utilize encoding or fragmentation of intermediate router's address. The encoding of each and every packet of course degrades the routing performance while fragmentation of address in small chunks may lead to state explosion problem that is discussed in [7]. As a result, none of the packet marking traceback techniques has been adapted for the practical work or implementation so far. In our previous work, we presented a PPM algorithm for IPv6 environment which is discussed in Section 3.

2.2 ICMP Traceback

ICMP traceback [1] scheme lies under the messaging category. Every router on the network is configured to pick a packet statistically (1 in every 20,000 packets

recommended) and generate an ICMP traceback message or iTrace directed to the same destination as the selected packet. The iTrace message itself consists of the next and previous hop information, and a timestamp. As many bytes of the traced packet as possible are also copied in the payload of iTrace. The time to live (TTL) field is set to 255, and is then used to identify the actual path of the attack.

This scheme can also be deployed on IPv6 networks and presents a very expandable technology if implemented with encryption and key distribution schemes. However, the additional traffic generated consumes a lot of bandwidth even with very low frequency (1/20,000). Without encryption, an attacker can inject false ICMP tra ceback messages. In addition, ICMP traffic is filtered in many organization to avoid several attack scenarios which make iTrace not that much useful.

2.3 Hash Based IP Traceback

It comes under packet digesting technique. In Hash-based traceback [5][6], officially called Source Path Isolation Engine(SPIE), specialized router confines partial information of every packet that passes through them in the form of hash, to be able in the future to determine if that packet passed through it. In this scheme such routers are called data generation agents (DGAs). DGA functionality is implemented on the routers. The network is logically divided into regions. In every region SPIE Collection and Reduction Agents (SCARs) connect to all DGAs, and are able to query them for necessary information. The SPIE Traceback Manager (STM) is a central management unit that communicates to IDSs of the victims and SCARs.

This technique is very effective and capable of identifying a single packet source as well as, according to best of our knowledge, the only scheme that also has solution for IPv6 networks [8]. This scheme, on the other hand, is very computational and resource intensive because tens of thousands of packets can traverse a router every second, the digested data can grow quickly to an enormous size, which is especially problematic for high-speed links.

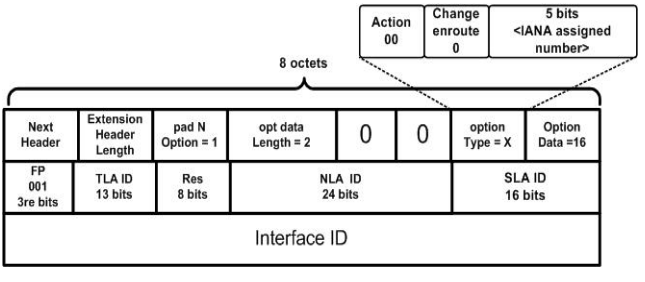


Fig. 1. Proposed Marking Field

3 Probabilistic Packet Marking (PPM) for IPv6 Traceback

This section will briefly discuss our previous work i.e. PPM for IPv6. In PPM for IPv6, router en route probabilistically marks the incoming packets with the Global unicast IPv6 address of that router. We used Hop-by-Hop Header [16] to store a mark

the reasons were two folds; first, the Hop-by-Hop option is processed by every router en-route. Second, it provides the larger space to store a mark. Proposed option in Hop by hop option header is shown in Figure 1.

Use of extension headers gave us great flexibility to pass the information to the victim. As we marked the packet with complete address, our scheme is not vulnerable to state explosion problem [7]. We used these marked packets to construct the reverse routing table from victim to attackers. For this purpose, on victim side, we proposed a data structure called Reverse Lookup Table (RLT). Following steps were taken to complete the traceback.

1. The victim will sort the RLT by distance field; as shown in figure 2.
2. Observe the discontinuity in distance field and apply the error correction algorithm (ECA) to find the missing nodes.
3. Finally, victim will resolve the last hop field to complete the RLT.

The resultant sorted tuples of routers can provide a complete path from Victim to attacker.

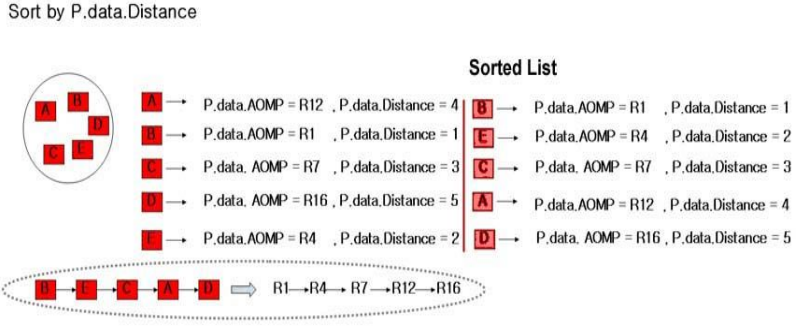


Fig. 2. Reconstructed path using AOMP value and Distance value

This algorithm worked under the assumption that victim is in DDoS attack so the number of evading packets would be sufficient to provide the information of all routes. However, it is quite practical the victim does not have complete route information of the attacker. For this purpose, we also introduced the Error Correction Algorithm [17]. Marking the packet with extra 20 bytes might increases the size of packet than PMTU, and since intermediate routers cannot do fragmentation, the packets will be dropped. Therefore, we also proposed a modified Path MTU (PMTU) discovery algorithm discussed in detail in [17].

4 Policy Based IP Traceback (PBIT)

4.1 Motivation of Another Traceback Technique

Thousands of packets traverse through one router in a second and marking of every packet, even probabilistically, may affect routing performance. Therefore, the

cooperation in implementing the traceback algorithm will not be tempting for ISPs. Because it is obvious, none of the ISP provides security to other networks by sacrificing their own customers' satisfaction. To cope with these problems, there should be a mechanism to minimize the burden of packet marking and initiate packet marking only when a victim is under (D)DoS attack.

One of the ways to accomplish this is to deploy IDS on victim side and once this IDS detects an attack it sends message to intermediate routers to initiate marking. However, since we do not have any information of path (because we are not using PPM here that is discussed above) we cannot send the message to desired routers to start marking. The other option left is to multicast the message to all backbone routers that is quite impractical due to many reason such as increase in network traffic that may lead to network congestion. Moreover, if going along with standards, we will have to use ICMP to send these messages and ICMP traffic is mainly filtered in many ISPs. Therefore, there are much greater chances that these messages will be dropped by most of the ISPs.

Another possible way is that IDSs are deployed on intermediate routers and starts marking packets, once they detect congestion or high packet rate on any specific interface. This scheme seems appealing by keeping in mind that most of the routers now come with IDS functionality or we may plug-in the IDS functionality in a router as a separate module (if this feature is present in router). The problem with this architecture that these types of router or routers with IDS are normally deployed on the edges of network due to the fact that adding IDS support to backbone routers will degrade the routing performance as IDS requires high end processing to infer something about attacks.

4.2 PBIT Mechanism

To mitigate the above problems we utilized the power of Policy Based Management System [12]. Policy-based management is an administrative approach that is used to simplify the management of a given endeavor by establishing policies to deal with situations that are likely to occur. The description of Policy Based Management is out of scope of this paper but it would be worthy to mention two basic building blocks of Policy Based Management architecture i.e. Policy Decision Point (PDP) and Policy Enforcement Point (PEP). PDP is a resource manager or policy server that is accountable for handling events and making decisions based on those events (for instance; at time *t* do *x*), and updating the PEP configuration appropriately. While the PEP exists in network nodes such as hosts, routers and firewall. It enforces the policies based on the "if condition then action" rule sets given by the PDP. Both PDP and PEP communicates with each other through COPS (Common Open Policy Service) that is a typical protocol [12], although DIAMETER or even SNMP may be used.

To go with policy based management framework, of course due to standard, we slightly modified our architecture. Instead of probabilistically marking of every packet by intermediate routers, we maintain a list of participating edge routers (the router closest to the sender) on PDP and placed an IDS along with traceback agent near to the victim as shown in Fig. 3.

Once the IDS detects a (D)DoS attack on victim, it generates the request to PDP to enforce policy which in turns, send message to all participating routers (i.e. PEP) found in the list to initiate packet marking *deterministically*. Most of the IDSs detect an attack after observing a huge traffic volume, and if we start probabilistic packet marking after this point, we might not have large amount of marked packets to construct the complete path. Therefore, in PBIT, we deterministically mark the packets so one packet would be enough to get the entire path. Actually, through this algorithm, we are not getting the entire path of an attack instead; we will be able to get only the injection point of an attack but finding the address of an ingress point is as good as full path traceback.

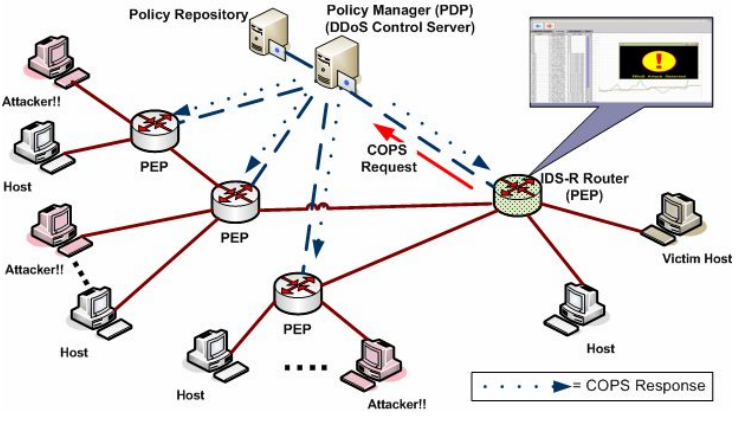


Fig. 3. Network architecture of policy based traceback system

The foremost improvement through this modification is obviously the lesser burden on intermediate routers of marking packets even they are not under (D)DoS attack hence will not affect the routing performance. Moreover, by using COPS, we are not deviating ourselves from standards else we could have a specialized server which maintains the list of participating routers and signal them to start packet marking after getting an indication from IDS. The complete pseudo code of PBIT is given below.

At source:

```
M = max (PMTU, 1280) - 26 bytes;
for every packet p{
    if p.size > M{
          fp[i]=fragment(p,M);
                send(fp[i]);
          }else
                send(p);
    }
```

At edge routers (PEP):

```
Marking procedure at edge router R:
Let attack is set to 1 when R got a signal from PDP:
    for every packet p{
          if (attack=1)
                mark_packet(p);
          forward(p);
    }
```

At Victim:

For traffic logging:

```
          for every marked packet pm
          if (pm.interface_addr is in RLT)
              incr_packetcount(if_addr,current_time);
             else{
              add_in_RLT(if_addr);
              set_packet_count(if_addr,1,current_time);
             }
```

For Traceback:

```
          If packet qm is given
           If_addr=Get_ifaddrr(qm);
          Else
           If_addrr=max_count(RLT,time_period);
```

5 Implementation and Evaluation

In this paper, we presented both of our architectures for IPv6 traceback i.e. PPM and PBIT. In case of PPM, we were interested in the number of packets required to get the full path to the victim. For this, we developed a simulator in Java, as there is no good support for IPv6 networks in current network simulators. On the other hand, the efficiency of PBIT depends on the IDS that how accurately and quickly it detects an attack. For PBIT evaluation, we integrated our traceback agent to IDS as shown in Fig. 4 developed by our lab. The performance of this IDS system has already been evaluated in [11].

Below we are comparing the efficiency of implemented scheme with key evaluation metrics discussed in [1]; this paper gave several evaluation metrics but here we are judging our scheme to only those attributes that can be affected by proposed algorithm. The detail comparison is shown in Table 1.

Processing Overhead: The processing can take place either at victim side or at intermediate nodes. For an ideal traceback scheme, the processing overhead of traceback should be minimum. Although the Figure 4 represents the traceback agent as an integrated part but in fact it is acting as a separate component. Therefore, in PBIT the processing overhead at intermediate nodes and victim side is almost none. Although during traceback intermediate nodes will consume a little processing power

to mark a packet however, this kind of processing can be seen in *Time To Live (TTL)* and *Hop Limit* calculations in IPv4 and IPv6 networks respectively. Furthermore; it is apparent; the proposed scheme does not require any calculation of hash values or message digests, encoding/decoding or any other computational intensive job either on intermediate routers or at victim side.

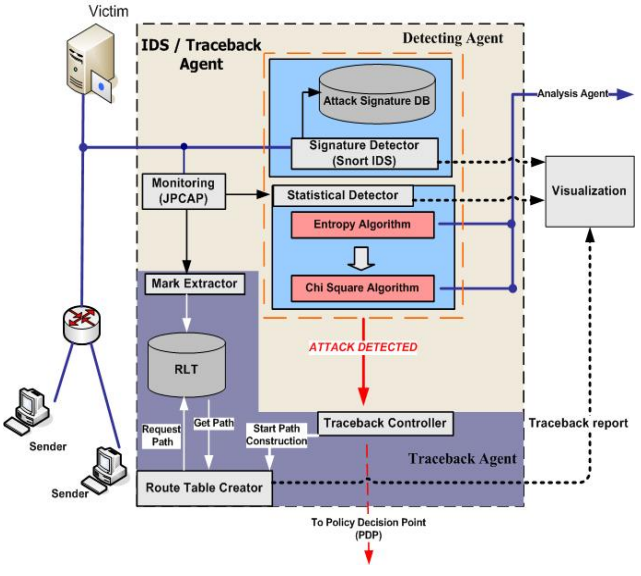


Fig. 4. Block diagram of overall architecture on victim side

Number of Attacking Packets: In PBIT, after (D)DoS attack detection, only one packet is enough to complete traceback which also eliminates the path reconstruction problem; one of the major weakness of PPM techniques.

ISP Involvement: As discussed above the traceback scheme should be tempting enough for ISPs because none of the ISP will compromise on quality of service and provide accountability of its user to other ISPs. If you ponder, you may realize that this is the motivation of PBIT. If any of the edge routers is not participating in traceback it can sincerely inform others ISPs or an ISP can also examine by observing the absence of the traceback option in this case the ISP which is not implementing PBIT would be considered as potential hacker and marking should be implemented on the interface(s) connected to that client ISP. It is pertinent to mention that for other IP traceback mechanism if intermediate nodes don't participate than it's nearly impossible to trace back an attack path.

Bandwidth Overhead: During traceback, we might need to slightly compromise on bandwidth consumption due to addition of one option header but this is acceptable as we already have much bigger routing header in IPv6 specification.

Table 1. Comparison of PBIT with other Traceback schemes

		iTrace	Hash-based	PPM	PBIT
Number of attacking packets		Thousands	1	Thousands	**1**
ISP involvement		Low	Fair	Low	**Low**
Network processing overhead	Every packet	Low	Low	Low	**None**
	During Traceback	None	Low	None	**Low**
Victim processing overhead	Every packet	None	None	None	**None**†
	During Traceback	High	None	High	**Low**
Bandwidth overhead	Every packet	Low	None	None	**None**
	During Traceback	None	Low	None	**Very Low**
Memory requirement	Network	Low	Fair	None	**None**
	Victim	High	None	High	**Low**
Ease of Evasion		High	Low	Low	**Low**
Protection		High	Fair	High	**High**
Can handle attacks other then DDoS		No	Yes	No	**No**

Ease of Evasion: Refers how easily an attacker can circumvent the traceback technique. In the case of PBIT we assume that edge routers are not compromised. For such instances, PPM algorithm will work best due to its distributed nature.

Protection: Relates to produce the meaningful traces if some of the devices included in traceback are undermined. PBIT is highly protective as intermediate routers don't participate in traceback and the single point of consideration is the router interface closest to the attacker if this interface or a router is down then there would be no way for an attacker to invade.

6 Conclusion

In this paper, we gave an introduction of IP traceback and a brief overview of current IP traceback trends. These schemes were not adapted widely for IPv4 networks. One of the main reasons was degradation in routing performance, as encoding should be applied to pass the path information through a limited space IPv4 header.

In this paper, we discussed two Packet Marking algorithms for IPv6 network. The extension header gave us great flexibility to pass the path information to the victim

† Considering IDS as an external component.

and since in both of our algorithms, information of routers are not distributed in different fragments as proposed in [3], our schemes are not affected by the state explosion problem that is discussed in [7]. We believe that PBIT is more appealing than PPM as it requires minimum ISP intervention and doesn't harm the routing performance. However, in the case of PBIT we assume that edge routers are not compromised. For such instances, PPM algorithm will work best due to its distributed nature.

References:

[1] Belenky, A. and Ansari, N. "On IP Traceback," IEEE Communications Magazine, Volume 41, Issue 7, July 2003
[2] S. Savage et al., "Network Support for IP Traceback," IEEE/ACM Trans. Net., vol. 9, no. 3, June 2001, pp. 226-37.
[3] Dawn X. Song and Adrian Perrig, "Advanced and authenticated marking schemes for IP traceback," in Proceedings IEEE Infocomm 2001, April 2001
[4] K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," Tech. Rep. CSD-00-013, Department of Computer Sciences, Purdue University, June 2000.
[5] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer. Single-packet IP traceback. ACM/IEEE Transactions on Networking, Dec.2002.
[6] Aljifri, H. "IP traceback: a new denial-of-service deterrent" Security & Privacy Magazine, IEEE , Volume: 1 , Issue: 3 , May-June 2003 Pages : 24 - 31
[7] Marcel Waldvogel, "GOSSIB vs. IP Traceback Rumors", 18th Annual Computer Security Applications Conference (ACSAC '02).
[8] W. Timothy Strayer, Christine E. Jones, Fabrice Tchakountio, and Regina Rosales Hain, SPIE-IPv6: Single IPv6 Packet Traceback, Local Computer Networks, 2004. 29th Annual IEEE International Conference on 16-18 Nov. 2004 Page(s):118 – 125.
[9] Micah Adler, "Tradeoffs in probabilistic packet marking for IP traceback," in Proceedings of 34th ACM Symposium on Theory of Computing (STOC), 2002.
[10] A. Belenky and N. Ansari, .On IP traceback,. IEEE Communications Magazine, vol. 41, no. 7, July 2003.
[11] Choong Seon Hong , Pil Yong Park, Wei Jiang, " DDoS Attack Defense Architecture Using Statistical Mechanism on Active Network Environment ", Applied Cryptography and Network Security , pp. 47-56, June 2004
[12] A. Westerinen et al, "Terminology for Policy-Based Management", RFC3198, IETF, November 2001.
[13] http://en.wikipedia.org/wiki/Intrusion-detection_system
[14] http://en.wikipedia.org/wiki/Intrusion_prevention_system
[15] http://en.wikipedia.org/wiki/Firewall_%28networking%29
[16] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF, December 1998.
[17] Syed Obaid Amin, Myung Su Kang and Choong Seon Hong, "A Lightweight IP Traceback Mechanism on IPv6", EUC Workshops 2006, LNCS 4097, pp. 671 – 680, 2006.

An Efficient Authentication and Simplified Certificate Status Management for Personal Area Networks*

Chul Sur[1] and Kyung Hyune Rhee[2]

[1] Department of Computer Science, Pukyong National University,
599-1, Daeyeon3-Dong, Nam-Gu, Busan 608-737, Republic of Korea
kahlil@pknu.ac.kr
[2] Division of Electronic, Computer and Telecommunication Engineering,
Pukyong National University
khrhee@pknu.ac.kr

Abstract. Recently the concept of personal PKI was introduced to describe a public key infrastructure specifically designed to support the distribution of public keys in a personal area network. However, traditional public key signature schemes and certificate status management schemes used in the personal PKI concept cause formidable overheads to components in the personal area network since mobile devices constituting the personal area network have limited computational and communication capabilities. In this paper we propose an efficient authentication protocol that eliminates the traditional public key operations on mobile devices without any assistance of a signature server. Moreover, the proposed protocol provides a simplified procedure for certificate status management to alleviate communication and computational costs on mobile devices in the personal area network.

1 Introduction

A Personal Area Network (PAN) is the interconnection of fixed, portable, or moving components within a range of an individual operating space, typically within a range of 10 meters. In PAN the communication between components should be secure and authenticated since private information and personal data will be transmitted over radio links. Secure and authenticated communication can be achieved by means of proper security protocols and appropriate security associations among PAN components.

For the sake of supporting key management in a PAN, a personal CA, which is responsible for generating public key certificates for all mobile devices within

* This work was partially supported by grant No. R01-2006-000-10260-0 from the Basic Research Program of the Korea Science & Engineering Foundation, and the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 273–282, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the PAN, was introduced in [3]. The personal CA is used by an ordinary user at home or small office deployment distinguished from large scale or global CA functions. Nevertheless, in order to use a personal PKI technology as like a conventional PKI technology, this concept assumes that at least one device in the PAN acts as a personal CA so as to issue certificates and provide certificate status management to all other devices. Therefore, all the personal devices can be equipped with certificates issued by the same CA, i.e., the personal CA, while sharing a common root public key. As a result, mobile devices in the PAN can establish secure and authenticated communications with each other by means of certificates. The initialization phase of [3] extends the concept of imprinting [12] to bootstrap all mobile devices with public key certificates. After all the mobile devices have been imprinted with their public key certificates, mobile devices may launch routine operations of the PAN by means of the traditional public key signature schemes.

The personal PKI concept seems to be properly applied to PAN environment. However, the personal PKI concept leaves at least two important challenging problems unaddressed. The first challenging problem to think about is that: The traditional public key signature schemes put resource-constrained mobile devices to formidable workloads since a digital signature is a computationally complex operation. The second challenging problem is that: To manage certificate status information, no optimization was devised and the conventional certificate status management schemes were considered. Consequently, to design efficient authentication protocol and certificate status management that addresses the aforementioned problems is a promising challenge for PAN environment.

In this paper, we propose an efficient authentication protocol that reduces computational overheads for generating and verifying signatures on mobile devices. Especially, we focus on eliminating the traditional public key operations on mobile devices by means of one-time signature scheme, and we differentiate it from previously proposed server-assisted computation approaches relied on assistances of a signature server. As a result, the proposed protocol gets rid of inherent drawbacks of server-assisted computation approaches such as problematic disputes, and high computational and storage requirements on a server side. Moreover, our protocol provides simplified certificate status management based on hash chain technique to alleviate communication and computational costs for checking certificate status information.

2 Preliminaries

2.1 One-Time Signatures and Fractal Merkle Tree Traveral

One-time signature (OTS for short) schemes are digital signature mechanisms which can be used to sign, at most, one message[7]. One-time signature schemes have the advantages that signature generation and verification are very efficient, and further, more secure since these schemes are only based on one-way functions, as opposed to trapdoor functions that are used in traditional public key signature schemes.

Despite of aforementioned advantages, one-time signature schemes have been considered to be impractical due to two main reasons: (1) the signature size is relatively long in comparison with traditional public key signatures. (2) their "one-timed-ness", i.e., key generation is required for each usage, thus implying that the public key must be distributed in an authentic fashion which is done most typically using a public key signature. As a result, the benefit of usefulness of quick and efficient one-way function is apparently lost.

In order to decrease the length of one-time signatures, a message digest should be calculated using hash function, just like traditional public key signature schemes, and then one-time signature scheme should be applied to the message digest. In this case, if a message digest would be the SHA-1 function which has a 160 bits output, we need 168 secrets to sign the message digest[7].

Merkle introduced the idea of using a hash tree to authenticate a large number of one-time signatures[8]. An important notion in Merkle's proposal is that of an *authentication path*: the values of all the nodes that are siblings of nodes on the path between a given leaf and the root. In [5], Jakobsson *et al.* presented *a fractal Merkle trees for the sequential traversal* of such a Merkle hash tree, which provides the authentication path for each leaf when the leaves are used one after the other. In this scheme, the total space required is bounded by $1.5log^2N/loglogN$ hash values, and the worst-case computational effort is $2logN/loglogN$ hash function evaluations per output. Recently, Naor *et al.* showed that the combined scheme of Merkle's one-time signature scheme with Jakobsson *et al.*'s algorithm provides fast signature times with low signature sizes and storage requirements through experimental results[9].

2.2 Hash Chain

The idea of *hash chain* concept is based on the property of a one-way hash function $h()$ that operates on arbitrary-length inputs to produce a fixed length value. One-way hash functions can be recursively applied to an input string. The notation $h^n(x)$ denotes the result of applying $h()$ n times recursively to an input x. That is,

$$h^n(x) = \underbrace{h(h(h(\ldots h}_{n\ times}(x)\ldots)))$$

Such recursive application results in a *hash chain* that is generated from the original input string:

$$h^0(x) = x, h^1(x), \ldots, h^n(x)$$

In most of the hash chain applications, first $h^n(x)$ is securely distributed and then the elements of the hash chain is spent one by one starting form $h^{n-1}(x)$ and continuing until the value of x is reached.

2.3 Modified Efficient Public Key Framework

In [13], Zhou *et al.* proposed a new public-key framework, in which the maximum lifetime of a certificate is divided into short periods and the certificate

could expire at the end of any period under the control of the certificate owner. They intended to establish a new public key framework that exempts the CA from testifying the validity of a certificate, once the certificate has been issued by the CA. However, Zhou's framework has considerable problems for practical implementation. That is, it is an unreasonable framework to authenticate an unidentified user based on some information submitted by the unidentified user in exempting CA. In particular, a malicious user can always generate valid signatures without any restriction. To overcome this drawback, they introduced a new trust party called security server. However, the security server is not only a redundant entity, but also requires an additional cost to be maintained securely.

Alternatively, we introduce *control window* mechanism to make Zhou's public-key framework above more suitable for realistic implementation.

Definition 1 (Control Window). Control Window *describes a time period that the verifier can trust the status of the sender's certificate only based on the sender's hash chain.*

Upon control window mechanism, CA sets the size of the control window of the user at the certificate issuance. The user can control the status of his/her certificate by using hash chain, and the verifier only trusts the user's hash chain during the control window. At the end point of the control window, the verifier queries certificate status information to CA.

3 System Model

3.1 Design Principles and Architecture

In this section, we firstly clarify our design principles in order to efficiently provide authentication and certificate status management among mobile devices in PAN environment. The concerns of our design are summarized as follows:

- *Eliminating Public Key Operations on Mobile Devices.* Since traditional public key signature schemes generally require computationally complex operations in terms of signature generation and even verification, they may not even be appropriate for resource-constrained mobile devices in PAN, which may have 8-bit or 16-bit microcontrollers running at very low CPU speeds. Therefore, designing an authentication protocol which does not perform any public key operations is a promising challenge in PAN environment.
- *No Assistance of a Signature Server.* To avoid cumbersome public key signature generations, some cryptographic protocols which depend upon a signature server were presented[1][2]. However, these approaches put a heavy burden on the server side or, both the server and the mobile device side in terms of high storage requirement for resolving problematic disputes. Furthermore, these approaches do not eliminate public key operation on verifier side and suffer from round-trip delay since all signing procedures are carried out through the signature server. Consequently, it is desirable to design an authentication protocol without assistances of the signature server.

– *Small Computational and Communication Overheads for Validating Certificate Status.* Although online certificate status checking mechanism such as OCSP[11] seems a good choice since mobile devices can retrieve timely certificate status information with moderate resources usages in comparison with CRLs[4], the personal CA suffers from heavy communication workloads as well as computational overheads as it requires computing lots of signatures. Therefore, to mitigate the personal CA's workloads, it is necessary to reduce the number of the personal CA's signature generations and total communication passes.

To define architectural model more clearly, we assume the followings:

– A PAN consists of portable or moving components that communicate with each other via wireless interfaces.
– At the constituting the PAN, all the security associations required for making PAN routine operations secure are set up. That is, every mobile device equipped with the PAN is bootstrapped with these security quantities and certificates during the initial phase.

A PAN is composed of a personal CA and mobile devices in our system model. The descriptions of system components are as follows:

– **Personal CA:** Personal CA is a unique trusted third party in the PAN, and it has a display and a simple input device to give its commands. Also, it is permanently available online to provide all other PAN components with certificates and certificate status information.
– **Mobile Devices:** Components equipped with the PAN, which have networking capability and likely low computing power.

3.2 Notations

We use the following notations to describe the protocols:

– PCA, M : the identities of personal CA and mobile device, respectively.
– $h()$: a cryptographic secure one-way hash function.
– SK_X : a randomly chosen secret key of the mobile device X.
– $sk_X^{i,j}$: the secrets of each one-time signature of the mobile device X, where

$$sk_X^{i,j} = h(SK_X|i|j)$$

i is the signature number, j is the index of the secret, and $|$ is the concatenation of messages.
– $pk_X^{i,j} := h(sk_X^{i,j})$: the commitments for each $sk_X^{i,j}$.
– $PLC_X^i := h(pk_X^{i,1}|\cdots|pk_X^{i,t})$: the i-th public leaf commitment, which is the hash of all the commitments of a single one-time signature.
– PK_X : a public key of the mobile device X, which is the tree root of a fractal Merkle hash tree.
– $AuthPath_X^i$: the authentication path of the i-th public leaf commitment of the mobile device X.

– VK_X^{n-i} : the i-th validation key of the mobile device X. Based on a randomly chosen secret quantity VK_X from the range of $h()$, the mobile device X computes the hash chain $VK_X^0, VK_X^1, \cdots, VK_X^n$, where

$$VK_X^0 = VK_X, VK_X^i = h_X^i(VK_X) = h_X(VK_X^{i-1})$$

VK_X^n constitutes X's *root validation key*, VK_X^{n-i} is X's *current validation key*.
– Sig_X^i : the i-th one-time signature of the mobile device X.
– $Cert_X$: a certificate of the mobile device X.

4 Proposed Protocol

In this section, we present an efficient authentication protocol that provides fast signature generation and verification without any assistance of a signature server, and offers simplified certificate status checking by means of control window mechanism.

Initialization. The initialization of mobile devices is the modified version of manual authentication protocol[3] that inherently settles key distribution problem in one-time signature scheme. The detailed steps are as follows:

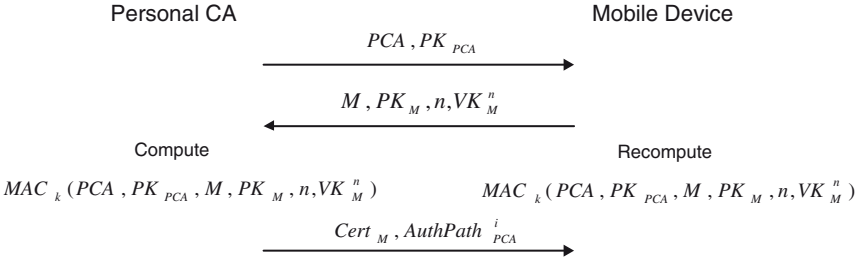


Fig. 1. System Initialization

1. The personal CA sends its identifier and public key to a mobile device.
2. The mobile device randomly generates two secret quantities SK_M and VK_M. Starting with these values, the mobile device performs the followings:
   – Generates the one-time secrets/commitments pairs and the corresponding public leaf commitments according to the total number of signatures n (Taking into account the PAN environment, we assume that the total number of signature is less than 2^{16}).
   – Initializes a fractal Merkle hash tree of height $logn$, and computes a public key PK_M, with the public leaf commitments values PLC_M^i as its leaves, where $i = 1, \cdots, n$.
   – Generates $VK_M^n = h^n(VK_M)$ as the root validation key.
   – Sets the signature number $i = 0$.

   Then, the mobile device submits M, PK_M, n, VK_M^n to the personal CA.

3. Both the personal CA and the mobile device carry out the following manual authentication:
   - The personal CA generates a random key k and computes a MAC as a function of $PCA, PK_{PCA}, M, PK_M, n, VK_M^n$ by using the random key k. The MAC and the key k are then displayed by the personal CA.
   - The user now types MAC and k into the mobile device, which uses k to recompute MAC value (using its stored versions of the public keys and associated data as input).

   If two values agree then the mobile device gives a success signal to the user. Otherwise it gives a failure signal.
4. If the mobile device emits a success indication, the user instructs the personal CA to generate a certificate. In order to generate the certificate, the personal CA sets up a control window CW according to the system security policy and issues the certificate signed by one-time signature for the mobile device together with the authentication path $AuthPath_{PCA}^i$ of the certificate.

$$Cert_M = \{Ser\#, M, PK_M, n, VK_M^n, CW, Sig_{PCA}^i\},$$

   where $Ser\#$ is a serial number.
5. The mobile device checks the followings to verify the correctness of the issued certificate.
   - Verifies the one-time signature of the personal CA on the certificate by the use of PK_{PCA} and $AuthPath_{PCA}^i$.
   - Checks whether the data fields within the certificate are valid as expected.

   If all the checks are successful, the protocol is now completed.

As described above, every mobile device is bootstrapped with a pair of public/secret key and its own certificate during the initial phase. After all mobile devices have been imprinted with their security quantities, mobile devices which wish to sign and verify a message carry out the following signature generation/verification phase.

Signature Generation. A mobile device M_s which wishes to sign a message m performs the followings:

- Proceeds Merkle's one-time signature scheme[7] as follows:
  - Increments the signature number i
  - Calculates a message digest $md = h(m)$ and sets C = number of '0'-bits in md, and then sets $msg = md||C$.
  - Generates $\{sk_{M_s}^{i,j}\}_{j=1}^t$ and the corresponding $\{pk_{M_s}^{i,j}\}_{j=1}^t$, where $t = |msg|$.
  - Calculates $Sig_{M_s}^i = \{sk_{M_s}^{i,j} \forall j \in \{j|msg_j = 1\},\ pk_{M_s}^{i,j} \forall j \in \{j|msg_j = 0\}\}$.
- Computes $AuthPath_{M_s}^i$ and updates authentication path using fractal Merkle tree algorithm[5].
- Calculates the current validation key $VK_{M_s}^{n-i}$.

Then, sends $m, Sig_{M_s}^i, AuthPath_{M_s}^i$ together with the signature counter i and the current validation key $VK_{M_s}^{n-i}$ to an intended mobile device M_v.

Signature Verification. The mobile device M_v proceeds the followings to check the status of the mobile device M_s:

- Obtains $Cert_{M_s}$ and queries whether the status of $Cert_{M_s}$ is valid or not.
- Verifies the current validation key based on the root validation key in the obtained certificate, i.e., $h^i(VK_{M_s}^{n-i}) \stackrel{?}{=} VK_{M_s}^n$.
- If all the checks are successful, the mobile device M_v caches $Cert_{M_s}$ and sets the current local time as starting trust time and the ending trust time based on the control window in $Cert_{M_s}$.

To verify the received signature, the mobile device M_v performs as follows:

- Calculates a message digest $md' = h(m)$ and sets C' = number of '0'-bits in md', and then sets $msg' = md'||C'$.
- Sets $Sig'_{M_s} = Sig_{M_s}$ by denoting $Sig'_{M_s} = \{sig'_j\}_{j=1}^t$, where $t = |msg'|$ and updates $sig'_j \longleftarrow h(sig'_j), \forall j \in \{j|msg'_j = 1\}$, and then calculates $PLC^i_{M_s} = \{sig'_1|\cdots|sig'_t\}$.
- Iteratively hashes $PLC^i_{M_s}$ with $AuthPath^i_{M_s}$ and compares the result to PK_{M_s} in the certificate $Cert_{M_s}$.

In comparison with previously proposed server-assisted computation approaches [1][2] to reduce computational overheads on resource-constrained mobile devices, the proposed protocol does not need to perform any public key operations and employ any signature server at all.

Also, the verifier needs not to query the signer's certificate status information to the personal CA during the period of control window since the verifier trusts the signer's certificate based on the hash chain in the certificate up to the ending trust point. As a result, our protocol provides moderate communication and computational overheads for validating certificate status compared with OCSP[11] and CRLs[4].

5 Evaluations

In this section, we give evaluations of the proposed protocol in terms of the security and performance points of view.

Security Evaluations. To provide secure operations, it is necessary to prove the security of both one-time signature scheme and control window mechanism used in the proposed protocols. Clearly, we require that message digest hash function $h()$ is collision-resistant. Then, it is sufficient that: if the one-way hash function $h()$ used for committing secrets and hash operations in the Merkle's one-time signature scheme is a collision-resistant function which implies preimage-resistant, no signature for a message $m' \neq m$ can be forged.

Regarding the security of control window mechanism, it is obvious that: to forge the mobile device's current validation key corresponding to the i-th one-time signature, an adversary should compute on his own the $(n-i)$-th $h()$-inverse of the root validation key VK^n in the mobile device's certificate, which is computationally infeasible work.

Performance Evaluations. Firstly, we compare the proposed protocol with the most efficient server-assisted computation approach[1] in terms of computational and storage requirements on system components. Note that computational requirement of our protocol for signer is comparable with [1] without putting a heavy burden on the server. Furthermore, signature verification of our protocol is more efficient than [1] since verifier does not perform traditional signature verification, but only needs to perform one-way hash computations. In particular, we solve the main problem in [1], which is the high storage requirement on the server by removing the signature server. Recall server-assisted computation approaches must store all signatures for resolving problematic disputes. In addition, considering storage requirement on the signer, our protocol only requires 1.9 KB approximately. (two 20 bytes security quantities and 1920 bytes hash tree along with 4 bytes signature counter) while [1] requires about 3.3 KB ($168 * 20$ bytes = 3.3 KB approximately.)

Upon taking into consideration of the efficiency of control window mechanism, clearly our protocol reduces the number of signature generations and communication passes of the personal CA since the verifier does not query certificate status information to the personal CA during the period of control window. To have concrete and general measurements in terms of communication costs, we consider the following parameters[10] to compare communication costs with CRLs and OCSP:

- n : Estimated total number of certificates ($n = 300,000$).
- p : Estimated fraction of certificate that will be revoked prior to their expiration ($p = 0.1$).
- q : Estimated number of certificate status queries issued per day ($q = 300,000$).
- T : Number of updates per day ($T = 2$, the periodic update occurs per 12 hours).
- C : Size of control window in our protocol ($C = 2$, the length of control window is two days).
- l_{sn} : Number of bits needed to hold a certificate serial number ($l_{sn} = 20$).
- l_{sig}, l_{hash} : Length of signature and hash value ($l_{sig} = 1,024$,$l_{hash} = 160$).

Table 1. gives the estimated daily communication costs according to three certificate status management schemes. If we make use of control window mechanism instead of OCSP, then communication cost for certificate status management can be diminished by 65%.

Table 1. Comparisons of Daily Communication Costs

Scheme	Communication Cost (bits)
CRLs	1.803×10^{11}
OCSP	3.132×10^{8}
Our Proposal	2.046×10^{8}

CRLs daily cost: $T \cdot (p \cdot n \cdot l_{sn} + l_{sig}) + q \cdot (p \cdot n \cdot l_{sn} + l_{sig})$
OCSP daily cost: $q \cdot l_{sn} + q \cdot l_{sig}$
Our protocol daily cost: $\frac{q \cdot l_{sn}}{C} + \frac{q \cdot l_{sig}}{C} + q \cdot l_{hash}$

6 Conclusion

In this paper, we have proposed an efficient protocol to reduce a burden of computation for digital signature generation and verification on the PAN components, and simplify the procedure of certificate status management in the PAN. Compared with sever-assisted computation approaches, the proposed protocol does not require performing any public key operations at all without assistances of a signature server. Based on hash chain technique, and further, the proposed protocol alleviates communication and computational costs for checking certificate status information.

References

1. K. Bicakci and N. Baykal, "Server assisted signature revisited," *Topics in Cryptology - CT-RSA 2003*, pp.143-156 March 2003.
2. X. Ding, D. Mazzocchi and G. Tsudik, "Experimenting with Server-Aided Signatures," *2002 Network and Distributed Systems Security Symposium (NDSS'02)*, February 2002.
3. C. Gehrmann, K. Nyberg and C. Mitchell, "The personal CA - PKI for a Personal Area Network," *Proceedings - IST Mobile & Wireless Communications Summit 2002*, June 2002.
4. R. Housley, W. Ford, W. Polk and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile," *RFC 2459*, January 1999.
5. M. Jakobsson, F. Leighton, S. Micali and M. Szydlo, "Fractal Merkel tree representation and traversal," *Topics in Cryptology - CT-RSA 2003*, pp.314-326, 2003.
6. L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, 24(11), 1981.
7. R. C. Merkle, "A digital signatures based on a conventional encryption function," *Advances in Cryptology - CRYPTO'87*, pp.369-378, 1987.
8. R. C. Merkle, "A certified digital signature," *Advances in Cryptology - CRYPTO'89*, pp.218-238, 1989
9. D. Naor, A. Shenhav and A. Wool, "One-Time Signature Revisited: Have They Become Practical?," Cryptology ePrint Archive, Report 2005/442, 2005.
10. M. Naor and K. Nissim, "Certificate revocation and certificate update," *The 7th USENIX Security Symposium*, January 1998.
11. M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, "X.509 Internet public key infrastructure on-line certificate status protocol (OCSP)," *RFC 2560*, June 1999.
12. F. Stajano and R. Anderson, "The resurrecting duckling: security issues for adhoc wireless networks," *The 7th International Workshop on Security Protocols*, pp.172-194, 1999.
13. J. Zhou, F. Fao and R. Deng, "An Efficient Public-Key Framework," *The 5th International Conference on Information and Communications Security*, pp.88-99, October 2003.

A Novel Rekey Management Scheme in Digital Broadcasting Network

Han-Seung Koo[1], Il-Kyoo Lee[2], Jae-Myung Kim[3], and Sung-Woong Ra[4]

[1] ETRI, Broadcasting System Research Group, Daejeon, Korea
koohs@etri.re.kr
[2] Division of I&C Engineering Dept, Kongju Univ., Chungnam, Korea
leeik@kongju.ac.kr
[3] Inha University, Inchun, Korea
jaekim@inha.ac.kr
[4] Dept. of EE, Chungnam National Univ., Daejeon, Korea
swra@cnu.ac.kr

Abstract. Conditional Access System (CAS) performs entitlement management to make only legitimate subscribers watch pay-services. Generally, CAS uses passive entitlement management to fulfill that entitlement control, and various schemes are existed for that. Among them, Tu introduced two schemes in [1], which are the simple scheme and complete scheme of four levels hierarchy. The advantage of the simple scheme of four levels hierarchy is a small key generation and encryption load for a CAS, but it is not good for the dynamic entitlement management. On the other hand, the complete scheme of four levels hierarchy is good for the dynamic entitlement management, but key generation and encryption load for CAS is considerable when it is compared to the simple scheme. In this circumstance, we proposed a novel scheme, which is an active entitlement key management. The proposed scheme not only performs the dynamic entitlement management very efficiently, but also generates and encrypts keys with a small load for CAS, which is just the same as the load of the simple scheme.

Keywords: Conditional Access System, Key Management, Digital TV Broadcasting System.

1 Introduction

Digital broadcasting system utilizes CAS with hierarchic key distribution model for access control. And 3 or 4 levels hierarchic key distribution model is a popularly used [1]-[6]. In case of 3 levels key distribution model, *control word* (CW), *authorization key* (AK), and *master private key* (MPK) are used [2], [5]. On the other hand, CW, AK, *receiving group key* (RGK), and MPK are used for 4 levels key distribution model. Note that, a disadvantage of 3 levels key distribution model compared to 4 levels one is a heavy system load in a key transmission point of view [1]-[3]. And *entitlement control message* (ECM) and *entitlement management message* (EMM) are used for delivering hierarchic keys [1]-[9].

CAS based on hierarchic key distribution model refreshes keys regularly and irregularly [6]. First of all, CAS refreshes keys regularly because it provides key

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 283 – 292, 2006.
© Springer-Verlag Berlin Heidelberg 2006

security and efficient billing. CAS performs efficient billing by synchronizing key refreshment period and service *charging time period* (CTP) [1]. However, since such frequent key refreshment causes a big system load, a trade-off between key security and frequent key refreshment is necessary. This regular key refreshment scheme is called *periodic entitlement management*. Second of all, CAS refreshes keys irregularly when extra key refreshment is necessary. For example, if a user wants to terminate his/her pay service or to change his/her current entitlement to another pay service before the entitlement is originally supposed to be expired, CAS performs irregular key refreshment. In this circumstances, CAS generally refreshes a key related to a channel or service group which a user wants to leave, and periodically sends refreshed keys to all users except the one who leave his/her entitlement. This irregular key refreshment scheme is called *non-periodic or dynamic entitlement management*. Note that CAS has to send keys periodically because all digital broadcasting standards [7]-[9] specifies one-way system as a mandatory requirement, and two-way system as an optional one. In other words, since CAS can't assure a reception of refreshed keys at a host's side in one-way system, there is no way but to send keys periodically for reliable key transmission. Unfortunately, this mechanism sometimes causes a big system load.

An existing solution for *periodic* and *dynamic entitlement management* has a big flaw when it is applied to a big system with tens or hundreds *pay-per-channel* (PPC) and hundreds of thousand or millions of subscribers. That is a heavy system load for key generation and encryption [1]-[4]. Especially in case of *dynamic entitlement management*, system load problem is getting more serious because a probability of occurring extra entitlement status change events definitely will goes up compared to a small system. This problem is what we resolved with the proposed scheme. With an active entitlement key management proposed in this paper, CAS can handle *periodic* and *dynamic entitlement management* with a small load and securely, even though a system is huge.

2 An Active Entitlement Key Management

In *passive entitlement management*, CAS refreshes keys and broadcasts them to a subscriber who leaves his/her entitlement. Note that a subscriber who leaves his/her entitlement is an old subscriber being deleted by system or any subscriber adapting or pausing his/her receiving group. However, in the proposed *active entitlement key management* (we will call this scheme as the active scheme in the rest of this paper), CAS just broadcast entitlement control information, *Authorization Revocation List* (ARL), including identifications of unauthorized subscribers, to subscribers when a subscriber leaves his/her entitlement. In other word, CAS with the active scheme doesn't need to refresh keys and broadcast them whenever dynamic entitlement management is necessary. But, since the active scheme should transmit additional ARL via EMM_{AK}, CAS requires more transmission bandwidth comparing to *passive entitlement management.* Therefore, we also propose the bandwidth efficient ARL transmission scheme by organizing ARL transmission table.

2.1 Key Hierarchy and Distribution Model

As shown in figure 1, the active scheme has four levels key hierarchy, such as MPK, RGK, AK, and CW. This key hierarchy model is exactly the same as the complete scheme, but the refreshment period of AK is not CTU, but CTP. In the complete scheme, it has to refresh AK per CTU to support *dynamic entitlement management* because it is based on *passive entitlement management* scheme. However, our proposed scheme broadcasts ARL to unauthorized subscribers to delete their invalid entitlement, so we don't need to refresh AK when a subscriber leaves his/her entitlement.

Head-end CA server broadcasts keys via EMM after generating them. In our scheme, there are two kinds of EMMs, which are EMM_{AK} and EMM_{RGK}. EMM_{AK} and EMM_{RGK} are used for delivering encrypted AK, i.e., $E_{RGK}\{AKs\}$, and RGK, i.e., $E_{MPK}\{RGK\}$, respectively. These messages have periodic broadcast frequency for reliable transmission of them. In case of EMM_{AK}, it is broadcasted per *AK retransmission period* (ARP), e.g., 0.1 ~ 15 seconds [2], and EMM_{RGK} is transmitted per CTP or when a subscriber subscribes new package service. Note that head-end CA server has to generate EMM_{AK} as many as the number of RG, and each EMM_{AK} is broadcasted to the corresponding RG. And there is no notation for CW and ECM in figure 1 because they are out of the scope of this paper. Additionally, our scheme provides the message authentication mechanism using MAC for EMM_{AK} because it contains ARL, and utilizes *EMM_{AK} authentication key* (EAK) for the authentication key of MAC. The details are described in the next section.

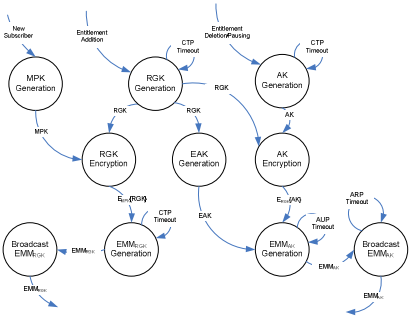


Fig. 1. Key Hierarchy and Distribution Model

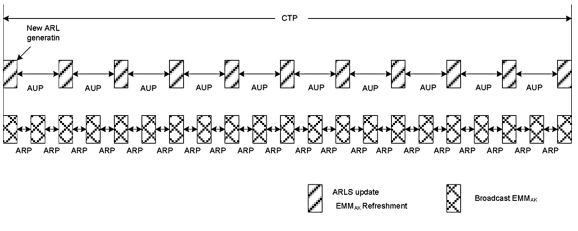


Fig. 2. Update and EMM_{AK} refreshment per AUP

2.2 Reliable Transmission of ARL

In this section, we describe the concept of ARL, the scheme of periodic transmission of ARL over EMM_{AK} for reliable transmission of ARL, and ARL authentication scheme with MAC algorithm.

2.2.1 Authentication Revocation List (ARL)

ARL is the list of the *record* which includes the identification code of an unauthorized subscriber. We can denote ARL as a group like {*record* 1, *record* 2, … , *record* M}, where M is the time variant number which varies according to the accumulated number of unauthorized subscribers during a CTP. Each RG has its own ARL, and head-end CA server generates N ARLs, where N is the number of RGs, then broadcasts them to the associated subscribers. We can define a group of ARLs as *ARL set* (ARLS) and denotes it as ARLS={arl_j|j=1,2,…,N}, where the arl_j is the ARL for the receiving group j. And we will use ARL_x notation in the rest of this paper for a arl_j of a certain receiving group. For example, at a subscriber's side which is included in ARL_x, after receiving ARL_x, *security processor* in ACU parses the ARL_x which is embedded in the EMM_{AK}, and verifies whether *records* in ARL_x include a matched identification with the ACU. If there is a matched *record* in ARL_x, the *security processor* deletes the entitlement stored in the memory immediately.

2.2.2 Periodic Transmission of ARLS over EMM_{AK}

One of the reliable transmission schemes of ARLS is periodic transmission of it via $EMM_{AK.}$ because there is no way to confirm whether subscribers receive ARLS correctly or not in a one-way digital TV broadcasting system. Note that a two-way digital TV broadcasting system is optional in all type of digital TV broadcasting systems [7]-[10].

The refreshment period of ARLS is CTP as shown in figure 2. Head-end CA server starts to generate new ARLS at every starting point of CTP and discards the old ARLS for previous CTP. And CA server updates ARLS only per AUP to reduce a system load for ARLS processing because subscribers leave their entitlement randomly during a CTP. Therefore, if CA server updates ARLS whenever subscribers leave their entitlement, it might cause a system load problem. Thus, in the proposed scheme, CA server temporarily stores incoming subscriber's leaving of his/her entitlement request and updates ARLS per AUP.

CAS takes advantage of EMM_{AK} for periodic transmission of ARLS because CAS originally retransmits EMM_{AK} per ARP, e.g., 0.1 ~ 15 seconds [2]. As a result, we can transmit ARLS periodically per ARP embedded in EMM_{AK}, and guarantee a reliable ARL transmission with a benefit of such a frequent retransmission period. Beside, we don't broadcast any messages specialized for ARLS, we can reduce the system complexity.

Despite the benefit of ARL update, there is a drawback of periodic ARLS broadcasting for the active scheme. That is an additional bandwidth consumption problem because ARLS is additional information when it is compared to *passive entitlement management*, its periodic transmission surly needs extra transmission bandwidth consumption. Therefore, we also proposed the efficient ARLS transmission scheme based on ARLS transmission table in 2.3, and describes the simulation result in chapter 3.

2.3 ARL Transmission Table

As described in previous section, CAS requires extra transmission bandwidth consumption in the active scheme. Therefore, we propose ARL transmission table to reduce the transmission bandwidth consumption efficiently.

Figure 3 shows the way to organize ARLS transmission table for ARL_x, i.e., one *receiving group* j. Note that head-end CA server should generate ARLS transmission tables for all element of ARLS, $\{arl_j | j=1, 2, \ldots, N\}$, where N is the number of RG, even though figure 3 depicts only for one of them.

There are two different ways of organizing the ARLS transmission table based on the size of ARL_x and the quotient of AUP divided by ARP, i.e., AUP/ARP. First of all, if the size of ARL_x is smaller than the AUP/ARP, head-end CA server organizes the table having a size of AUP/ARP by locating each *record*s of the ARL_x recursively (see figure 3.(a)). For example, if the value of AUP/ARP is 5 and the number of *records* in ARL_x is 2, then the ARLS transmission table for ARL_x will be $\{record_1, record_2, record_1, record_2, record_1\}$.

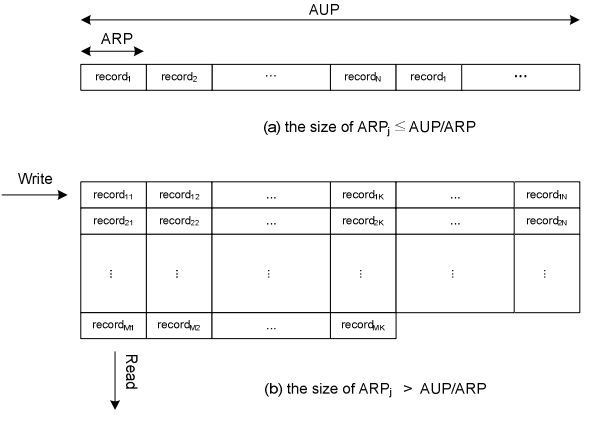


Fig. 3. Update and EMM_{AK} refreshment per AUP

Table 1. Example of Broadcasting ARL Transmission Table

ARP timeout order	Case 1	Case 2
1st ARP timeout	Send EMM_{AK} with $\{record_1\}$	Send EMM_{AK} with $\{ record_1, record_6\}$
2nd ARP timeout	Send EMM_{AK} with $\{record_2\}$	Send EMM_{AK} with $\{ record_2, record_7\}$
3rd ARP timeout	Send EMM_{AK} with $\{record_1\}$	Send EMM_{AK} with $\{ record_3\}$
4th ARP timeout	Send EMM_{AK} with $\{record_2\}$	Send EMM_{AK} with $\{ record_4\}$
5th ARP timeout	Send EMM_{AK} with $\{record_1\}$	Send EMM_{AK} with $\{ record_5\}$

Second of all, if the size of ARL_x is greater than the AUP/ARP, head-end CA server organizes the table having a shape of matrix like figure 3.(b), where N is the quotient of AUP/ARP. Note that the last row of the table might have fewer rows than N. For example, if the value of AUP/ARP is 5 and the number of *records* in ARL_x is 7, the ARLS transmission table for ARL_x will be

$$\begin{Bmatrix} record_1 & record_2 & record_3 & record_4 & record_5 \\ record_6 & record_7 & & & \end{Bmatrix}.$$

After the completion of organizing ARLS transmission table, CA server broadcasts the tables to subscribers by using two different ways according to the size of ARL_x. First, if the size of ARL_x is smaller than the AUP/ARP, head-end CA server broadcasts each element of the table at every ARP using the EMM_{AKj}. For example, if the ARLS transmission table is {$record_1$, $record_2$, $record_1$, $record_2$, $record_1$}, it is broadcasted like Table 1, case 1.

Second, if the size of ARL_x is greater than the AUP/ARP, head-end CA server broadcasts each column of the table at every ARP using the EMM_{AKj}. For example, if the ARLS transmission table is

$$\begin{Bmatrix} record_1 & record_2 & record_3 & record_4 & record_5 \\ record_6 & record_7 & & & \end{Bmatrix},$$

it is broadcasted like Table 1, case 2.

3 Performance Analysis

Table 2 shows the number of times of key generation based on the key refreshment frequency of the complete scheme and the active scheme with T channels, M charging groups, and N receiving groups. Note that, the value of M is the same as the number of days in a month [1], i.e., 30 days. First of all, the complete scheme generates AK T times per CTU because it refreshes AK per CTU, and T×M times per CTP. In case of RGK, the complete scheme generates RGK N+f(t) times. Here, N is the number of columns in a row of *the receiving group key matrix* [1], and f(t), $1 \le t \le M$, indicates the number of subscribers who leave his/her entitlement per CTU. Additionally, the complete scheme generates RGK per CTP $N \times M + \sum_{t=1}^{M} f(t)$ times with the value of M which indicates a total number of days in a month. On the other hand, the active scheme doesn't generate AK and RGK per CTU because it refreshes AK and RGK per CTP, and generates AK and RGK per CTP T and N times, respectively.

Figure 4 shows the simulation results of key generation load for a system with the assumptions like below.

1. Number of Subscribers (S) = one million, Number of Channels (T) = 100, Number of Receiving Groups (N) = 100, and Number of Charging Groups (M) = 30.
2. The number of subscribers who leave their entitlement is 1% of S, i.e., 10,000, and those leave events occur uniformly over a CTP. In other words, f(t) in Table 2 has a constant value of 10,000/M. at every CTU.

As shown in figure 4, the complete scheme has to generate AK and RGK about 533 (T+N+10,000/M) times per CTU, and about 16,000 (533×M) times per CTP. Besides, it is clear that as the number of subscribers who leave their entitlement increases, the complete scheme has to generate more keys than that the active scheme. On the other hand, the active scheme just generates AK and RGK 200 (T+N) times per CTP. Note that, CAS can generate them not only once in a CTP as shown in figure 4, but also

every each day by distributing the key generation load to every day's work. In this situation, if we choose second approach, CAS only has to generate keys about 7 times per CTU.

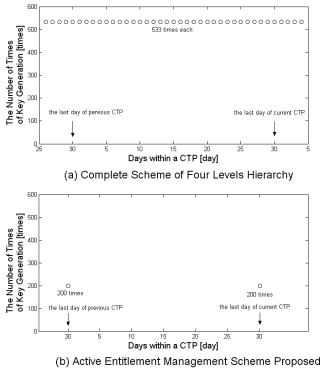


Fig. 4. The number of times of key generation

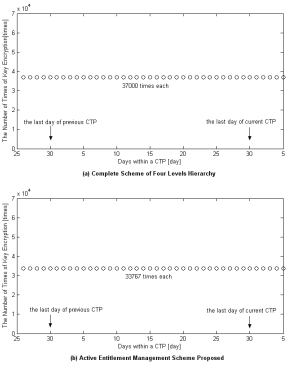


Fig. 5. The number of times of key encryption

Table 2. The number of times of key generation

	The complete scheme [1]	Active scheme proposed
AK per CTU	T	None
AK per CTP	T×M	T
RGK per CTU	N+f(t)	None
RGK per CTP	$N\times M+\sum_{t=1}^{M} f(t)$	N

Table 3 shows the number of times of key encryption of AK and RGK with S subscribers, N receiving groups, and M charging groups. First of all, the complete scheme has to encrypt AKs with RGK N×M times per CTU because there are N×M packages [1] to be broadcast per CTU. When we consider it for a CTP, the complete scheme has to encrypt AKS with RGK N×M×M times because CTP consists of M days. In case of RGK encryption with MPK in the complete scheme, CA system encrypts $S'(t)$ times per CTP, and $S'(t)$ consists of $S+\sum_{t=1}^{M} f(t)+\sum_{t=1}^{M} f'(t)$, $1\le t\le M$, here $f(t)$ indicates the number of subscribers who leave his/her entitlement per CTU and $f'(t)$ means the number of subscribers who add his/her entitlement per CTU. On the other hand, the active scheme doesn't need to encrypt AKs with RGK per CTU because it doesn't refresh AKs per CTU, and it encrypts AKs with RGK N times per CTP because there are N receiving groups. In case of RGK encryption with MPK in the active scheme, CA system encrypts $S(t)$ times per CTP, and $S(t)$ consists of $S+\sum_{t=1}^{M} f'(t)$, $1\le t\le M$, here $f'(t)$ means the number of subscribers who add his/her entitlement per CTU.

Table 3. The number of times of key encryption

	The complete scheme [1]	Active scheme proposed
$E_{RGK}\{AKs\}$ per CTU	N×M	N/A
$E_{RGK}\{AKs\}$ per CTP	N×M×M	N
$E_{MPK}\{RGK\}$ per CTP	$S'(t)$	S(t)

Figure 5 shows the simulation results of key encryption load for a system with the assumptions like below.

1. The same as the first and second assumptions in the key generation simulation.
2. The number of subscribers who adds their entitlement is 1% of S, i.e., 10,000, and those add events occur uniformly over a CTP. In other words, $f'(t)$ in Table 3 has a constant value of 10,000/M at every CTU.

With above assumptions, the complete scheme encrypts 37,000 (N×M + (S+ $\sum_{i=1}^{M} f(t) + \sum_{i=1}^{M} f'(t)$)/M) times per CTU, and 1,110,000 times per CTP. On the other hand, the active scheme encrypts about 33,767 ((N+ S+ $\sum_{i=1}^{M} f'(t)$)/M) times per CTU, and bout 1,013,010 times per CTP. As the simulation result shows, the active scheme encrypts about 3,233 times less per CTU, and about 96,990 times less per CTU. Note that if the number of subscribers who leave their entitlement increases, the complete scheme has to encrypt more keys than the case of active scheme.

Since the active scheme additionally broadcast ARLS periodically, extra transmission channel bandwidth consumption existed inevitably. However, the needed amount is just negligible, and we simulate it by calculating the average bit rate for the last AUP by varying AUP (see figure 6) to simulate it in the worst case. In other words, the size of ARLS will be a definitely maximum at that period, and this causes the greatest transmission channel consumption during a CTP. We assumed the size of *record* in an ARL and the value of ARP as 20 bits and 15 seconds, respectively. Note that the size of *record* in an ARL, 20 bits, is determined to make it possible to identify each subscriber among one million users, and the period of ARP,

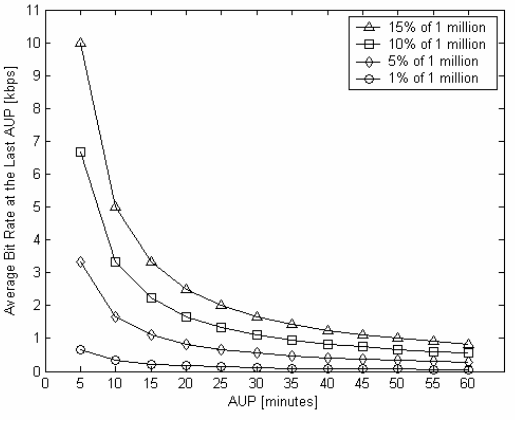


Fig. 6. Average bit rate for the last AUP by varying AUP

15 seconds, is selected among 0.1 ~ 15 seconds [2] to simulate in the worst circumstance. It is clear that we can see better result if we pick another ARP value, smaller than 15 seconds, eg., 0.1 seconds. As you can see in figure 6, if we choose 30 minutes for the AUP, and assume 1% of one million subscribers, 10,000, are leaving during a CTP, the necessary bandwidth for the transmission of ARLS is only about 0.11 kbps. Even though this result value could be increased as the number of subscribers who leave their entitlement goes up, the required bandwidth is very small, e.g., only about 1.67 kbps is necessary when we assume 15% of one million subscribers, 150,000, are leaving during a CTP.

4 Conclusion

In this paper, we proposed an active entitlement key management scheme for CAS on digital TV broadcasting system. We not only introduced a novel concept of ARL for dynamic entitlement management, but also designed key distribution model, including ARL, based on four levels key hierarchy, ARL authentication scheme for secure transmission of ARL, and ARL transmission table for efficient transmission bandwidth consumption. With the proposed scheme, we can reduce key generation and encryption load considerably compared to the complete scheme. Further, we can manage randomly changed users entitlement status securely and efficiently with the proposed scheme. We simulated this remarkable performance improvement by comparing the active scheme and the complete scheme with assumptions of one million subscribers, and one hundred PPC and receiving groups.

Acknowledgements

This research was supported by University IT Research Center Project (INHA UWB-ITRC), Korea.

References

1. F. K. Tu, C. S. Laih, and H. H. Tung, “On key distribution management for conditional access system on Pay-TV system,” *IEEE Trans. on Consumer Electronics*, Vol. 45, No. 1, Feb. 1999, pp. 151-158.
2. H. S. Cho, and S. H. Lee, “A new key management mechanism and performance improvement for conditional access system,” *Korea Information Processing Society Journal C,* Vol. 8, No. 1, Feb. 2001, pp. 75-87.
3. W. Lee, “Key distribution and management for conditional access system on DBS,” *Proc. of International Conference on Cryptology and Information Security,* 1996, pp. 82-86
4. T. Jiang, S. Zeng, and B. Lin, “Key distribution based on hierarchical access control for conditional access system in DTV broadcast,” *IEEE Trans. on Consumer Electronics,* Vol. 50, No. 1, Feb. 2004, pp. 225-230.
5. ITU Rec. 810, *Conditional Access Broadcasting Systems,* ITU-R, Geneva, Switzerland, 1992.

6. B. M. Macq, J-J and Quisquater, “Cryptology for digital TV broadcasting,” *Proceeding of the IEEE*, 1995, pp. 944-957.
7. ATSC Std. A/70A, *Conditional Access System for Terrestrial Broadcast, Revision A,* ATSC, Washington, D. C., 2004.
8. ETSI TS 103 197, *Head-end implementation of DVB SimulCrypt,* Sophia Antiplis Cedex, France, 2003.
9. ANSI/SCTE 40, *Digital Cable Network Interface Standard,* SCTE, Exton, PA, 2004.
10. H. S. Cho, and C. S. Lim, “DigiPass: Conditional access system for KoreaSat DBS, *The Institute of Electronics Engineers of Korea Society Papers,* Vol. 22, No. 7, July 1995, pp. 768-775

A New Encoding Approach Realizing High Security and High Performance Based on Double Common Encryption Using Static Keys and Dynamic Keys

Kiyoshi Yanagimoto, Takaaki Hasegawa, and Makoto Takano

NTT West Corporation
6-2-82 Shimaya, Konohana-ku
Osaka 554-0024, Japan
{k.yanagimoto, t.hasegawa, m.takano}@rdc.west.ntt.co.jp

Abstract. Recently, most of information systems such as customer management systems have been worked via networks. Encoding characters in databases or files is essential to prevent from leaking or stealing important information such as customer information. These systems, however, need the highest security level as well as higher performance. We propose a new method to improve performance without sacrificing security level. The key idea is encrypting only important information and using double common-key encryption with two types of keys, i.e. static keys and dynamic keys in order to shorten encrypting time and to secure systems. Our experiments revealed that high performance was realized at the same security level as a traditional way.

Keywords: Security, Network security, Database security, Common key encryption.

1 Introduction

Recently, most of information systems have worked via networks. In those systems, important information is transferred via networks. For example, users send their personal information to a customer management system on the Web in order to enjoy services. So, those systems need high security level in order to prevent from stealing, leaking or tampering with customer information. Encryption is essential to protect information through networks. Those systems also require high performance because many customers should be able to use those systems with no stresses. However, high security conflicts with high performance. As a system gets higher security, performance would be lower.

Our goal is to improve performance in network systems without sacrificing security. A network system consists of file servers and client terminals. We propose a new approach to encrypt information in network systems. We assume that not all information is important and need to encrypt. What is important can be defined in advance. It would take less time to encrypt only important information. On the other hands, we must consider keeping up security level. The key idea is encrypting only important information and using double common-key encryption with two types of

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 293 – 301, 2006.
© Springer-Verlag Berlin Heidelberg 2006

keys i.e. static common keys and dynamic common keys in order to shorten encrypting time and to secure systems.

The rest of this paper is organized as follows. We discuss prior work and their limitations in section 2. We propose a new approach for system security in section 3. Then, we describe experiments and evaluations in section 4, and discuss the approach in section 5. Finally, we conclude with future work.

2 Prior Work

There were two approaches in system security. One is network security and the other is database security. A lot of encrypting methods were proposed in network security. Network security were proposed and standardized at each layer of network, such as the application layer, the transport layer, the network layer and the data link layer [1, 2]. In those methods, payloads in each layer are encrypted. In high layers such as the application layer, only data are encrypted. In contrast, much information is encrypted in low layers. For example, IP addresses are encrypted in the network layer. So, encrypting in low layer has high security but it needs more machine resources. Encrypting payloads in low layers is secure but it would be inefficient since those methods encrypt not only important information but also non-important information. It also needs encrypting and decrypting at each server. Another limitation is that performance falls when encryption keys are changed every communication in order to secure data through network.

In database security, two popular methods were proposed. One is encrypting the all of a database. A disadvantage of this method is low performance because stored data need to be decrypted in data accesses. The other is encrypting fields in a database [3, 4]. This method is more efficient than encrypting all of a database. However, field indexing is still an open problem.

3 Double Common Encryption

3.1 Overview

We propose a new approach for security in networks and databases. Our approach is based on double encryption to only important information. We assume that not all information is important and need to encrypt. What is important information can be defined in advance. It would take less time to encrypt only important information. We must also consider keeping up security level. In order to realize it, we have two constraints. One constraint is to store encrypted data in a database. The other constraint is to encrypt the data so as to change the encrypted data through networks at each communication even if the same data are sent. We propose to use two types of common encryption keys. One is static common keys and the other is dynamic common keys. We assume that the static common keys are common to client terminals. We use the static common keys to store the encrypted data in a database. We propose to use a part of network information as the dynamic common keys. Network information is common to both a client terminal and a server, and it is

different at each communication. Since the dynamic common keys continually vary according to network information, we use the dynamic keys to change the encrypted data through networks at each communication. We propose double encryption using two types of keys. First, we encrypt important information by the static common keys. Then, we doubly encrypt the encrypted data by the dynamic common keys in order to secure the encrypted data through network moreover. The encrypted data does not need to be decrypted by the static common keys at the server, because we do not encrypt the data necessary to communication such as session information and we can store the encrypted data in a database as it is. So, performance would improve because only decryption by the dynamic common keys is needed at the server.

Our basic idea is as follows:

1. extracting important fields from a payload in a client terminal according to predefined information types and encrypting only the values in the important fields by the static common keys
2. encrypting the encrypted information by the dynamic common keys moreover and sending the doubly encrypted information to a server from the client terminal
3. decrypting the doubly encrypted information by the dynamic common key and stored it in a database (i.e. the stored information is still encrypted by the static common keys)

We show an example in Fig. 1. In step (1), the important data (abc) is extracted from the payload data (abc, def, ghi) at a client terminal. In step (2), the data is encrypted into the data (ikj) by the static common keys. In step (3), the data is moreover encrypted into the data (pos) by the dynamic common keys. In step (4), the payload data (pos, def, ghi) is sending to a server and the data (pos) is extracted. In step (5), the data (pos) is decrypted into the data (ikj) by the dynamic common keys. So, the encrypted data is stored in a database.

3.2 Encryption by Static Common Keys

Our proposing method needs to define a table in order to recognize important information. We show an example in Fig. 2. The table contains important information types as well as the static common encryption keys. It also contains IP addresses and HTTP (Hyper Text Transfer Protocol) port numbers of target host and origin host. Payloads containing important information are efficiently selected since IP headers are filtered according to IP addresses and port numbers. First, important information is extracted from a payload according to important information types in the table. We do not define the size of important information in the table since we can use identifiers in the payload in order to get the sizes of important information. Then, important information is encrypted by the static common keys in the table. Encryption is done by the byte so as to index encrypted information. In the upper row of this example shown in Fig.2, target IP address means the address of the server and origin IP address means the address of the client terminal. The filed “Name” is extracted from a payload and the value of this field is encrypted at the client terminal before sending to the server.

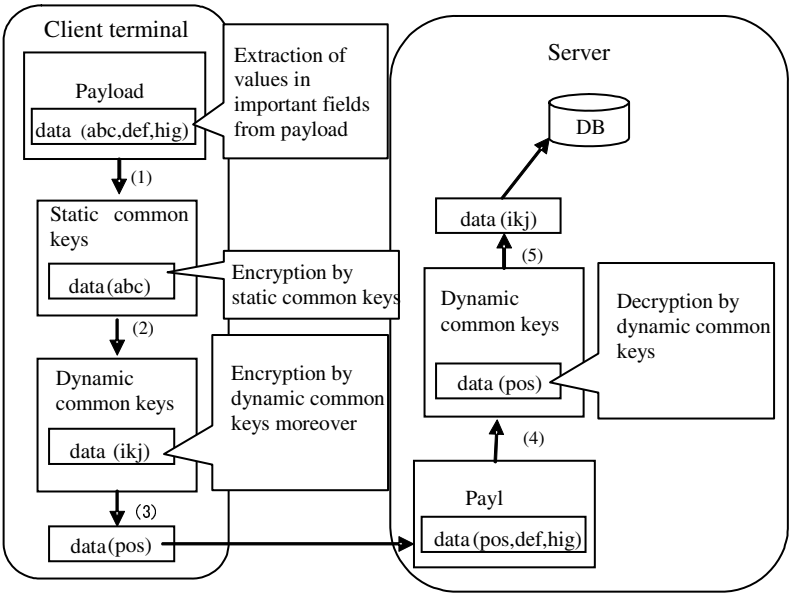


Fig. 1. Overview

3.3 Encryption by Dynamic Common keys

The encrypted information by the static common keys is doubly encrypted by the dynamic common keys at the client terminal before sending to the server. As the dynamic common keys, we use the identification of IP header in communication between the client terminal and the server. The dynamic common keys are variable since the identification of IP header is different at each communication. So the encrypted information is continually changed even if the original information is the same. The advantage of our method is that it does not need to exchange keys, since the identification of IP header is known to both the client terminals and the servers.

Prior work proposed to change keys so as to change encrypted information at each communication in order to raise security level. In order to keep the strength of the

Session information					Common encryption key	Direction of process	Important filed name
Target IP address	Origin IP address	Protocol	Target port	Origin port			
10.1.1.2	10.1.1.1	TCP	0	10000	XSDEDM	encryption	Name
10.1.1.1	10.1.1.2	TCP	10000	0	XSDEDM	decryption	Name

Fig. 2. Information for static common key encryption

encryption in network security, the sum of key length of a static common key and a dynamic common key is set to the same as prior work in network security. Even though the key length of a dynamic common key is enough short, the encrypted information varies at each communication and the strength of security is the same level as that of the prior work. So, another advantage of our method is that it would not take time as long as prior work.

3.4 Storage of Encrypted Information

The doubly encrypted information is sent from the client terminal to the server and then it is decrypted by the dynamic common keys at the server. The decrypted information by the dynamic common keys is stored to a database as it is. The decrypted information is secure because it is still encrypted by the static common keys. So, system performance never falls in database access, since our approach does not need to encrypt databases. Since encryption is done by the byte in our method, we can access the database rightly. If the static common keys are common among another client terminal, we can also access the database rightly by another client terminal.

4 Evaluation

We experimented with simulation based on designed systems. The system consists of a client terminal, load balancer, a file server, a database and networks. We simulate multi client access to a file server by using a load balancer. The system architecture of the proposing method and prior method are shown in Fig. 3. In prior method, first, the whole payload is encrypted at a client terminal, and then the encrypted payload is decrypted and encrypted again at the load balancer. Finally the whole encrypted payload is decrypted and extracted a field to data access. Note that the database is encrypted. On the other hand, encryptions by the static common keys and the dynamic common keys are done at a client terminal in our method. No encryptions are done at a load balancer. Only decryption by the dynamic common keys is done at a file server. Note that the database itself is not encrypted but important information is encrypted. We compare the time at each component in prior method and our method.

We set the network band to 100 Mbps, and we set the average packet size to 1,200 bytes. We assume that important information exists 500 bytes from the beginning of a payload and the size of important information is 100 byte. We used 56 bytes in DES (Data Encryption Standard) as prior method. In our method, we set the sum of the key length of a static key and a dynamic key to 56 bytes based on DES. 3 packets are needed in a request and a response to a database from a client terminal respectively. A client terminal frequently accessed to a file server up to 30 times. We used a relational database. Data structure is shown in Table 1. 128 bit key length in AES (Advanced Encryption Standard) was used as prior work in database security. Ten thousand records were stored in the database.

We compared performance of each system by timing at each component, i.e. a client terminal, a load balancer, and a file server. First, we evaluated each time in a client terminal. The results are shown in Table 2. The time of prior method was 0.201ms and the time of our method was 0.024ms. So, the results showed that

performance improved about 8 times. Second, we evaluated each time in a load balancer. In prior method, encrypted packets should be decrypted and encrypt again because cookie information in packets needs to be extract. Our method does not need to decrypt and encrypt at a load balancer since the cookie information were not encrypted. It took time to select a file server and send to the server according to cookie information. The time of prior method was 0.435ms and the time of our method was 0.267ms. So, the results showed that our method reduced 0.168ms and performance improved about 1.6 times. Finally, we evaluated each time in a file server. The time of prior method was 247.2ms and the time of our method was 155.0ms. So, the results showed that our method reduced 92.2ms and the performance improved about 1.6 times. At all, our method reduced 92.5ms and the performance improved about 1.6 times.

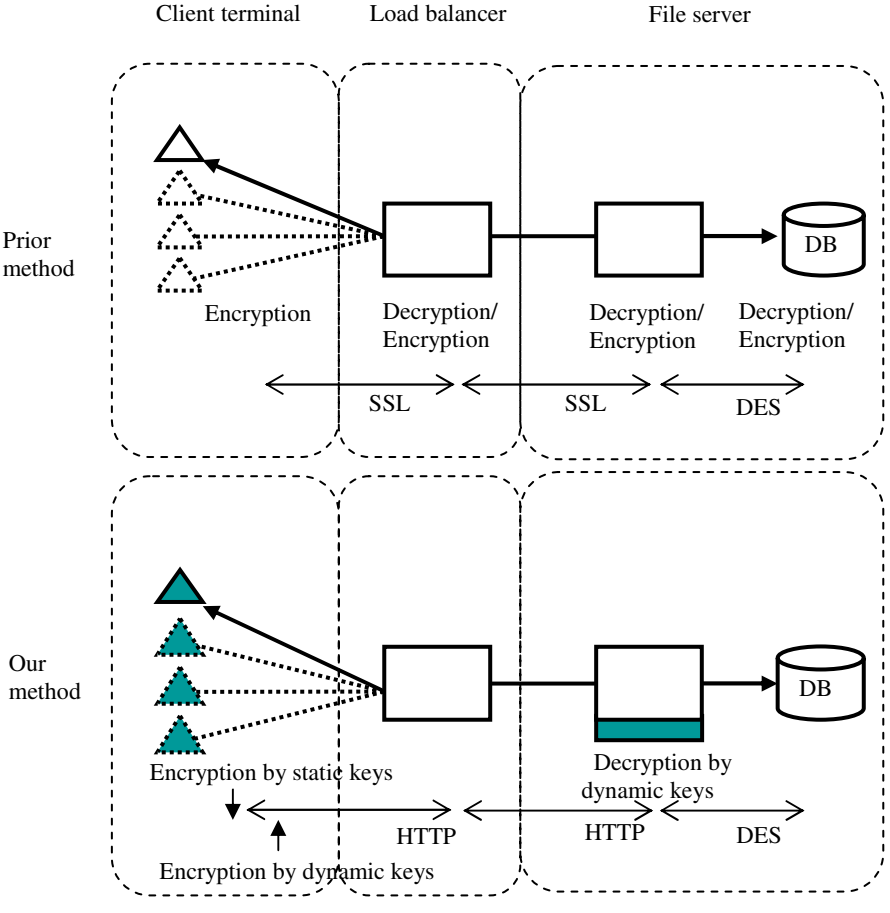


Fig. 3. System Architectures

Table 1. Data structure in experiments

Columns	Type	Size (byte)	Encryption
Name	varchar	100	yes
Address	varchar	300	no
Telephone	varchar	12	no
Credit card	int	15	no

Table 2. Evaluation of performance

Average transaction time	Prior method (ms)	Our method (ms)
Client terminal	0.201	0.024
Load balancer	0.435	0.267
File server	247.2	155.0
Total	247.8	155.3

5 Discussion

5.1 System Performance

Our experiments showed that the transaction time at a file server is dominant and our method successfully reduced the transaction time at a file server. We found data access time was dominant at a file server. So, we investigated the time in data access by varying amount of stored data in the database. We show the results in Table 3. When the amount of stored data increased, the time increased in prior methods, while performance does not fall in our method. Our method does not need an encrypted database while a prior method needs an encrypted database. The result said it took more time to access to an encrypted database when the amount of stored data increased. So, the effect of our method does not need an encrypted database without lowering security level. Our method also reduced the processing time at a client terminal and a load balancer respectively.

Table 3. Time in data access

Stored data (#record)	Prior method (second)	Our method (second)
1000	0.171	0.146
5000	0.204	0.153
10000	0.247	0.155

In addition, prior method needs to exchange keys at a client terminal and a load balancer. Our experiments showed it took extra 4.15ms to exchange keys. Our method does not need to exchange keys because it does not need to share the keys at anytime.

5.2 Strength of Security

Our experiments revealed that performance improved. Now we discuss the strength of security. We compared the strength of security in prior method and our method. The results are shown in Table 4. The encrypted block size is short in our method since the information needs to be encrypted by the byte. It is because we can access to the database without encrypting the database itself. However, the sum of the length of a static encryption key and the length of a dynamic encryption key is equal to the length of the encryption key in prior method. Our method also has an advantage of not needing to exchange encryption keys and not needing to keep them. So, the strength of encryption in our method would be the same level as in prior method.

5.3 Limitations

We discuss limitations of our method. One of the limitations of our method is multi client access to a database. While the advantage of our method is not to need to keep static encryption keys in the file server, our method forces to keep static encryption keys in the client terminal instead. So, if multi client terminal need to access the file server, the static common encryption keys must be shared among multi client

Table 4. Strength of security

	Prior method	Our method
Encryption block size (byte)	7 or 8	1
Key length in network communication	56	56
Key length in database (bit)	56	54
Need to exchange encryption keys	yes	no
Need to keep encryption keys	yes	no

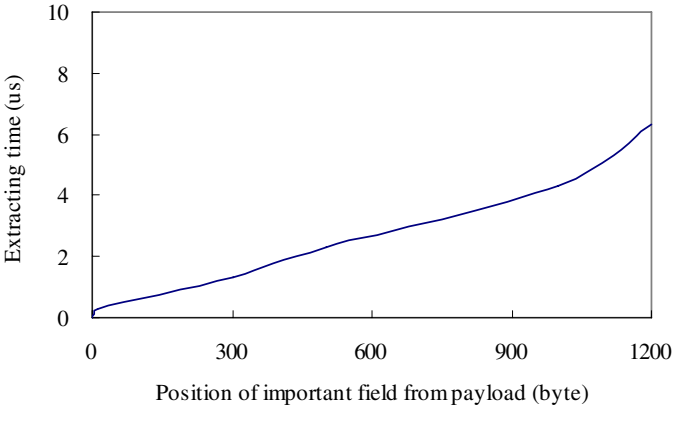


Fig. 4. Time in extraction important fields from a payload

terminals. Our future work is to research how the static encryption key can be shared in multi client terminals.

We did not investigate to extract an important field from a payload. As we simply extracted an important field from the beginning of a payload, the extracting time depended on the position of the payload. We show the extracting time according to the position of the payload in Fig. 4. The extraction time increased in proportion to the position of the payload. We have still room for improvement in extracting important fields.

6 Conclusion

We proposed a double common encrypting method to only important information. The key idea was extracting the values of important fields from a payload and encrypting doubly the values of important fields by static common keys and dynamic common keys. The most advantage was that our method did not need encrypt the database itself. The experiments using a network simulator revealed that the performance improved without falling security level. In the future, we are planning to improve our method by tuning parameters as well as to investigate the method for distributing the static common keys to multi client terminals when one of the client terminals accesses to a file server for the first time.

References

1. IEEE (Institute of Electrical and Electronic Engineers), “802.1AE - Media Access Control (MAC) Security”, http://www.ieee802. org/1/pages/802.1ae.html
2. IETF (Internet Engineering Task Force), “IETF Homepage”, http://www.ietf.org/
3. Oracle, “Transparent Data Encryption”, http://www.oracle.com/technology/oramag/oracle/05-sep/o55security.html
4. “eCipherGate”, Hitachi System and Services, http://www.hitachi-system.co.jp/eciphergate/ (In Japanese)

GMPLS-Based VPN Service to Realize End-to-End QoS and Resilient Paths

Hiroshi Matsuura[1] and Kazumasa Takami[2]

[1] NTT Service Integration Laboratories
9-11, Midori-Cho 3-Chome, Musashino-Shi, Tokyo 180-8585, Japan
matsuura.hiroshi@lab.ntt.co.jp
[2] Faculty of Engineering Soka University
k_takami@t.soka.ac.jp

Abstract. We propose hierarchically distributed path computation elements (HDPCEs) that manage a multilayered GMPLS network and provide traffic engineering paths for end-to-end routes. We apply HDPCEs to provide a VPN service on optical networks for IP customers. One of the VPN services that HDPCEs provide is individual-flow-based bandwidth allocation, which assures end-to-end QoS. However, conducting flow-based bandwidth assurance has been difficult because of the heavy burden on the admission controller (AC) and its routing delay. We made conducting flow-based bandwidth assurance possible by distributing ACs and advanced route creation. Another service provides resilient paths that are free from failure in the optical domain. There is no clear SLA on an individual path basis for the path backup policy. HDPCEs provide backup methods, such as protection, preplanned restoration, and active restoration according to the SLA with the customer.

Keywords: GMPLS, QoS, SLA, PCE, VPN.

1 Introduction

Generalized multiprotocol label switching (GMPLS) [1] is now being applied to multilayered networks that include IP and optical network domains. In particular, for the core domain of the GMPLS network, optical fibers are used, and they facilitate communication among IP domains that are connected to the core domain. In this situation, a customer in one IP domain can communicate with another customer via a virtual private network (VPN) on the optical core domain. From the VPN provider's point of view, assuring quality of service (QoS), security, and providing resilient paths for customers is important. We apply our previously proposed hierarchically distributed path computation elements (HDPCEs) [2] [3] to a VPN service and to achieve these requirements.

To achieve end-to-end QoS, bandwidth assurance for each flow in an IP domain and in a multidomain environment is indispensable. End-to-end QoS was assured mainly by applying Differentiated Services (Diffserv) [4] [5] and Bandwidth Broker (BB) because Diffserv aggregates many flows into a small number of Diffserv classes and reduces the control plane burden. On the other hand, Diffserv has difficulty

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 302 – 311, 2006.
© Springer-Verlag Berlin Heidelberg 2006

assuring QoS on an individual-flow basis. HDPCEs conduct individual-flow-based call admission control and determine an end-to-end multidomain route for each flow. At the same time, HDPCEs have a function to assure required bandwidth of the flow; thus, there is no need to use BB. Appropriate routes in each domain are calculated in advance and maintained in the route list in the corresponding HDPCE; thus, each routing burden can be minimized.

To provide robust security to each customer, maintaining the secrecy of customer information, including link-state information from other customer domains, is important. If the legacy OSPF routing protocol is applied to interdomain route determination, link-state information of an IP domain is required by the PCE in the source domain because the PCE has to conduct multidomain routing. Therefore, maintaining link-state information within a domain is difficult. On the contrary, interdomain HDPCE, which conducts route determination in the interdomain, eliminates the need for link-state flooding among IP domains. Therefore, link-state information in one IP domain is kept secret from other domains.

Legacy PCEs do not have a backup function that provides an appropriate backup strategy for each end-to-end path of a customer. HDPCEs provide each customer with three different backup methods: protection, preplanned restoration, and active restoration depending on the service level agreement (SLA) between a VPN provider and its customer.

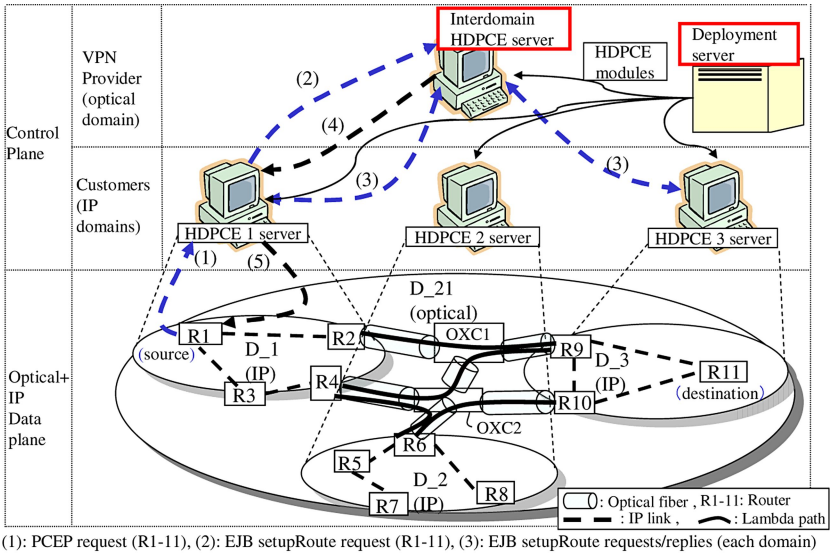
In Section 2, we describe the HDPCE deployment architecture and a procedure of creating an interdomain route using HDPCEs. We also show the SLA terms that are agreed upon between a VPN provider and a customer who uses the VPN service. In Sections 3 and 4, we show how HDPCEs provide end-to-end QoS and backup methods to IP customers depending on the SLA with them. Section 5 concludes our paper.

2 HDPCE Architecture

First, the authorized HDPCE deployment server distributes an HDPCE software module to HDPCE servers using the function of online Enterprise Java Bean (EJB) [6] deployment. HDPCE modules are distributed to the interdomain HDPCE server, which manages the VPN provider's optical interdomain, and to the customer HDPCE server, which is used for the customer IP domain. An example of this distribution is shown in Fig. 1. After the distribution of the HDPCEs, each network operator registers domain routers, optical cross connecters (OXCs), links between the routers and OXCs, initial bandwidths, and costs of links in the distributed HDPCE. After the registration of these links, the interdomain HDPCE establishes optical lambda paths through optical fibers in the optical interdomain. The HDPCE can find the shortest route for the lambda path from one domain to another using the interdomain shortest path first (SPF) algorithm [2] [3], and bandwidths reserved between two domains are determined by the SLA with the customer.

After the establishment of necessary lambda paths for the customer, the VPN provider allows the customer to communicate with other IP domains via the VPN. An example of establishing an end-to-end IP path by using HDPCEs is shown in Fig. 1. In this example, we suppose that source IP router R1 in IP domain D_1 asks HDPCE

1 to determine the route from R1 to R11 in IP domain D_3 by a PCEP request [7]. In this case, HDPCE 1 judges that there is no R11 in D_1 and forwards the request to the interdomain HDPCE. The interdomain HDPCE 21 selects the appropriate lambda path between D_1 and D_3 from its interdomain route list. Once the lambda path whose route is R2-OXC1-R9 is chosen, the interdomain HDPCE delegates the underlying routing to the underlying HDPCE 1 and HDPCE 3 specifying border routers R2 and R9, respectively. The underlying HDPCEs choose the best routes in their domains from their route lists. Finally, the optimal end-to-end route is chosen. This route is sent back to source router R1 in the PCEP reply message [7]. Among HDPCEs, RMI over IIOP [8] protocol is used for the interaction of EJBs.



(1): PCEP request (R1-11), (2): EJB setupRoute request (R1-11), (3): EJB setupRoute requests/replies (each domain)
(4): EJB setupRoute reply (R1-11), (5): PCEP reply (R1-11)

Fig. 1. HDPCEs deployment and their interdomain routing

SLA

VPN provider

Traffic	Amount of individual traffic from customer IP domain to other IP domains via VPN.		
End-to-end routing policy	Non-revelation	No link-state information is revealed from the customer IP domain.	
	Partial-revelation	Cost weight	From the customer IP domain, BVLs, which are virtual links between BRs, are revealed to interdomain HDPCE.
	Full-revelation	Cost weight	From the customer IP domain, BVLs and IVLs, which are virtual links between BR and IR, are revealed to interdomain HDPCE.
Backup policy	Protection	In the optical interdomain, a customer-specified path is backed up by a 1:1 protection path which is SRLG-disjointed from the path.	
	Preplanned restoration	In the optical interdomain, a customer-specified path is backed up by a N:1 restoration path which is SRLG-disjointed from the path.	
	Active restoration	In the optical interdomain, once the customer-specified path malfunctions, an alternate path is dynamically chosen.	

Customer

Fig. 2. SLA terms between a VPN provider and a customer

As shown in Fig. 2, SLA terms are determined between a VPN provider and its customer. Besides the traffic from the customer domain to other domains, there are two main policies in SLA.

For the end-to-end routing policy, the customer can chose three options. The first option is the "nonrevelation" policy, which reveals no virtual links (VLs) from a customer's IP domain. Thus, interdomain route-selection is conducted without using link states in the customer IP domain, only the costs of interdomain lambda paths are considered. This policy is the cheapest among the three options because the burden of interdomain HDPCE is the lightest and the interdomain HDPCE can use effective lambda paths for the optical domain without consideration of the customer IP domain.

The second option is the "partial-revelation" policy, which reveals border virtual links (BVLs), which are VLs between two border routers (BRs) in the customer IP domain. It is more expensive than the "nonrevelation" policy because an interdomain HDPCE takes account of the VLs from the domain in addition to the interdomain lambda paths. Applying this policy to a domain that performs the role of a hub for other domains is very useful. For example, once a customer has three IP domains that are connected by the VPN and one of the IP domains is mainly used for transit between the other two domains, setting a "partial-revelation" policy in the transit domain is a good strategy.

The third option is a "full-revelation" policy, which reveals BVLs and inner virtual links (IVLs), which are VLs between an inner router (IR) and a BR. It is the most expensive among the three options because the interdomain HDPCE creates all possible routes from all registered IRs and BRs in the customer IP domain to other domains and from other domains to all IRs and BRs in the domain.

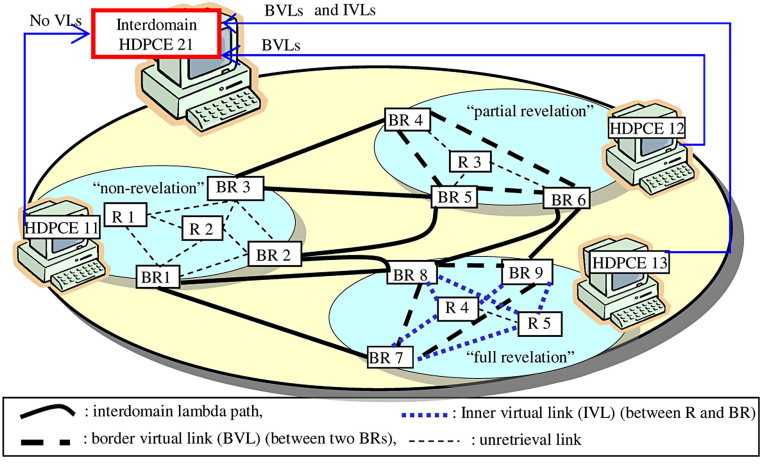


Fig. 3. Underlying route information retrieval performed by an interdomain HDPCE

An example in which a VPN provider has three customer IP domains that have these three policies is shown in Fig. 3. These retrievals of VLs from underlying HDPCEs are conducted when the interdomain HDPCE renews its interdomain route

list. An underlying HDPCE copies the corresponding route cost in its route list as the cost of a VL when it sends VLs to the interdomain HDPCE.

In the "partial-revelation" and "full-revelation" policies, the parameter "cost weight" should also be determined in the SLA. As shown in Eq. 1, the sum of lambda path costs "$lambda_{cost}$"s and VL costs "VL_{cost}"s of lambda paths and VLs, which are on the interdomain route, is the cost of the interdomain route "$Route_{cost}$".

$$Route_{\cos t} = x \sum_{lambda \in route} lambda_{\cos t} + \sum_{VL \in route} y(VL_{\cos t}) \quad (1)$$

Parameter x is the cost weight of "$lambda_{cost}$"s and y is the cost weight of "VL_{cost}"s, and the value of y is varied depending on the SLA contract with each customer. Therefore, if cost weight y of the VLs in a customer domain is larger than those of other IP domains, the domain is not likely to be used for the transit domain because $Route_{cost}$ which passes through the domain is more expensive than other route costs. Therefore, this high-cost-weight method is advantageous for the customer for reserving link capacity.

There are three options in the SLA for backup policies.

The first option is protection that provides 1:1 protection for the primary path. This option imposes a minimum delay in the case of switching paths from primary to protection because streaming data for the primary path is copied to the protection path in the optical domain. On the other hand, this option is expensive because 1:1 protection requires ample bandwidth in the optical network. In this option, 1:1 protection means each primary path is protected by one protection path.

The second option is preplanned restoration in which multiple primary paths share one restoration path. Even though this option does not assure recovery from simultaneous failures in multiple primary paths, it is cheaper than the protection option because the restoration path is shared.

The third option is active restoration, which does not prepare the backup path beforehand and search the restoration path when a malfunction occurs in the customer's path. This option is the cheapest among the three options, even though it does not assure the backup path and has routing overhead when the path is switched to the restoration path.

For each customer, the path backup policy can be determined on an individual-path basis; therefore, protection, or preplanned restoration policy is likely to be applied to important paths. We discuss these backup policies further in Section 4.

3 End-to-End QoS Management Performed by HDPCEs

Legacy QoS management systems mainly combine DiffServ with Bandwidth Broker (BB), though conducting flow-based QoS management is difficult. On the other hand, legacy PCEs [9] [10] try to determine each flow-based end-to-end explicit route by traffic engineering based on the OSPF protocol. The OSPF protocol is, however, designed for hop-by-hop routing and is not necessarily suitable for PCE-based multidomain source routing, especially for a GMPLS-based VPN service.

For example, in the OSPF protocol, link states of a domain are not usually visible from other domains; therefore, assuring end-to-end QoS is difficult. If one domain reveals its link states to other IP domains using OSPF virtual links, secret information

such as the link state of a domain is not kept secret with respect to other domains. To make matters worse, optical interdomain link states of a VPN provider are revealed to IP domains as forwarding adjacency (FA) LSPs; thus, the VPN provider cannot assure security.

In addition, in OSPF-based legacy PCEs, Dijkstra's shortest path first (SPF) algorithm [11] runs whenever a PCEP setup route request comes, and it calculates the shortest path from the source router to the destination router. The source domain PCE, however, has an overhead of calculating the shortest path, especially if there are many domains and links involved. In addition, in end-to-end QoS, Dijkstra's SPF algorithm does not check the available bandwidth of each link, so it requires the help of BB to check whether the path has sufficient bandwidth to satisfy the QoS of each application flow. BB, however, only checks the bandwidth of the path and does not propose the alternative path that satisfies the bandwidth requirement.

HDPCEs resolve these problems of the OSPF protocol, and each flow-based QoS is assured. First, HDPCEs enable the removal of link-state information flooding among domains including the optical interdomain. This is because, as shown in Fig. 1, each HDPCE manages the link states in its own domain and updates the link costs and remaining bandwidth of each link after the establishment of a path in the link. VLs are revealed by an HDPCE of an IP domain to the interdomain HCPCE, but the revelation is restricted to the VPN provider and never revealed to other customers.

Routing delay is minimized because each HDPCE calculates the shortest routes between possible source and destination routers/domains in its own domain and updates these routes in its route list asynchronously with the route-setup request. Distributed and parallel processing of HDPCEs also contributes to lighten the burden of routing and increase the speed of the routing. In the case shown in Fig. 1, parallel processing is conducted by HDPCE 1 and HDPCE 3 for the route selection in D_1 and D_3, respectively. If the number of VPN customers increases, dividing the interdomain HDPCE further and preventing the concentration of the burden on one interdomain HDPCE is also possible. For example, as shown in Fig. 1, if three new customers join the VPN service, allocating these three customer domains under another interdomain HDPCE and placing the inter-interdomain HDPCE on top of the two interdomain HDPCEs is possible.

As shown in Fig. 1, an interdomain route is determined in a hierarchical manner. In Fig. 4, we explain route-selection flow details in an HDPCE. As shown in Step 3, an HDPCE chooses the route on a route-cost basis and on an available bandwidth basis from the routes in its route list. After the determination of the route in the domain, the HDPCE updates the remaining bandwidth and cost of the links on the route, as shown in Step 4. Therefore, the probability of finding the path that satisfies the bandwidth requirement becomes greater compared with that of the combination of the OSPF protocol and BB.

The SLA of the end-to-end routing policy, "full revelation policy," has the highest probability of accommodating the most paths that satisfy the required bandwidth in the customer domain. This is because the cost of a VL is determined by the least-loaded (LL) algorithm [12], where the cost is defined as the inverse of the remaining bandwidth. Therefore, VLs that have less remaining bandwidth are not likely to be used. On the other hand, links in a customer domain that has a "nonrevelation policy" are likely to be used without consideration of remaining bandwidth; thus,

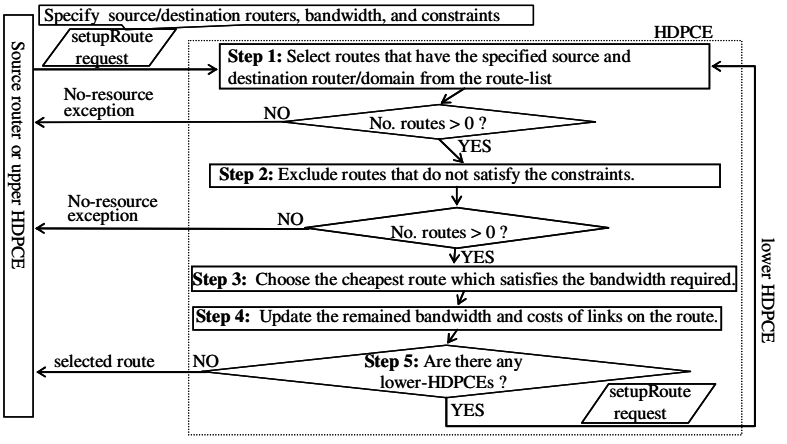


Fig. 4. A route selection flow in an HDPCE

QoS-assured end-to-end routes are less likely to be created than those of other customer domains.

4 Backup Strategies Provided by HDPCEs

The interdomain HDPCE for the optical domain of the VPN provider offers three methods for backing up individual IP paths based on the SLA with the customer who owns these paths.

A shared risk link group (SRLG) [13], whose unit is mainly an optical fiber or a conduit of fibers, is introduced to define the group of links that share the risk of failure

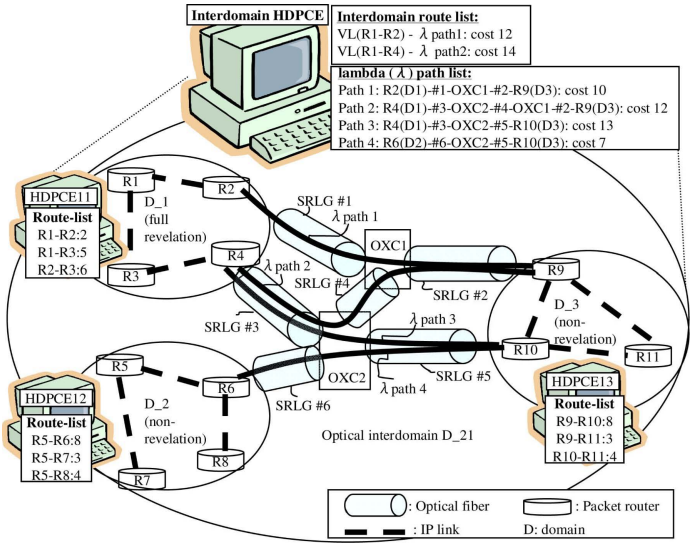


Fig. 5. SRLG management by interdomain HDPCE

with other group members. In legacy OSPF-based routing, however, selecting an SRLG disjointed from the primary path is difficult because Dijkstra's SPF algorithm does not take SRLG information into account and only considers link costs [14].

As is shown in Fig. 5, interdomain HDPCE establishes lambda paths based on the required traffic among IP customer domains and the interdomain route-costs. Each lambda path is related to its component SRLG sequence and its cost in the lambda path list. Sometimes a lambda path passes through one or more than one IP domains, which are transit domains. $lambda_{cost}$ is defined as

$$lambda_{\cos t}(n, B, \alpha, \beta) = \alpha\beta / B + (100 - \alpha)n \,, \tag{2}$$

where n indicates the number of route components of the lambda path such as OXCs and optical fibers, B indicates the available bandwidth of the lambda path, α is used to weight of the inverse of B against n within the range between 0 and 100. β is used for compensating between the inverse of B and n before the IP paths are accommodated in the lambda path. On the other hand, α is flexibly changed depending on the conditions of the applied interdomain. For example, if the number of disjointed SRLG pairs between any two IP domains becomes limited, setting α to a small value for effective accommodation of the IP paths in the lambda paths is better. That is because selecting a lambda path that has a small number of route components for a primary IP path helps to select a protection IP path, which is SRLG-disjointed from the primary path. On the other hand, for large networks that have many alternative disjointed SRLG routes between any two packet domains, setting α to a large value for effective accommodation of the IP paths in the lambda paths is better. That is because available bandwidth B is weighted more and balanced use of lambda paths is conducted.

A flowchart of how an interdomain HDPCE provides three different backup methods for individual IP paths is shown in Fig. 6. Step 1 shows the flow for the selection of a primary IP path, but if some optical fiber of the primary IP path has a

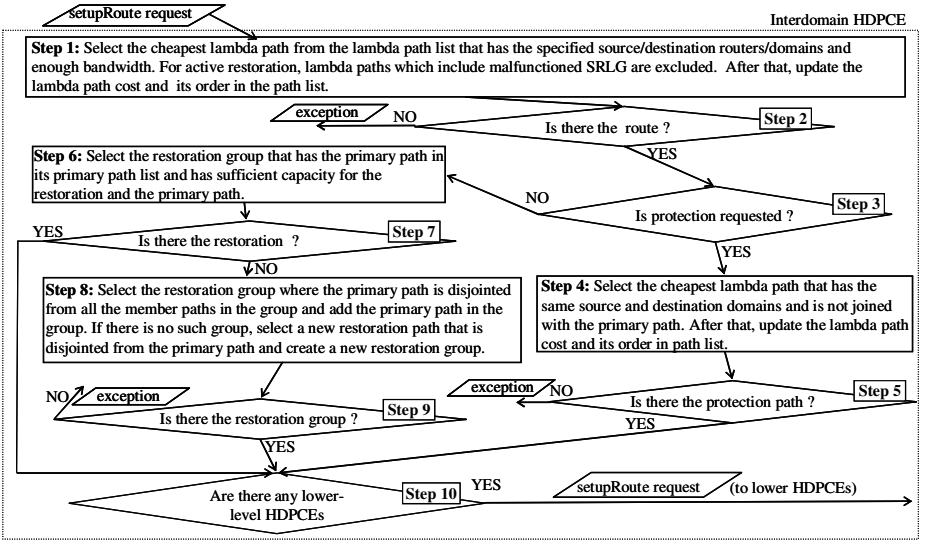


Fig. 6. Disjoint SRLG backup flowchart in an interdomain HDPCS

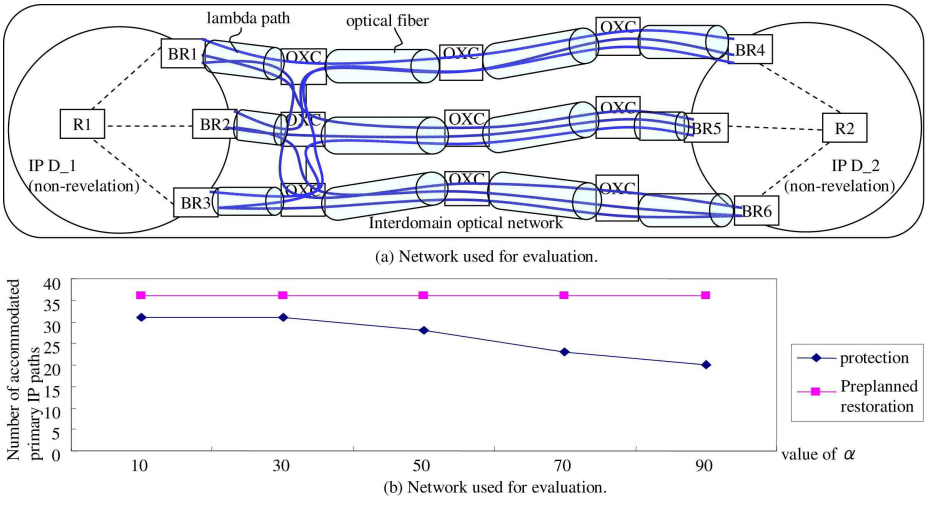


Fig. 7. Comparison between protection and preplanned restoration

malfunction, Step 1 is also used to select an active restoration path excluding the malfunctioning SRLG. The protection IP path for the primary IP path is selected in Step 4 in which a disjointed SRLG lambda path is chosen from the lambda path list to accommodate the protection IP path.

In addition to the lambda path list, the interdomain HDPCE maintains a list of restoration groups. A restoration group consists of SRLG-disjointed primary lambda paths and their shared restoration lambda path. In Step 6, the interdomain HDPCE searches the restoration group that includes the same lambda path in which the primary IP path is accommodated. If such a group exists, the restoration lambda path in the restoration group is used for the backup of primary IP path. If there is no such group, the primary lambda path is added to another restoration group, or a new restoration group is created in Step 8.

The GMPLS network shown in Fig. 7(a) is used for the evaluation of two proposed backup methods: protection and preplanned restoration methods. The network has an optical interdomain managed by an interdomain HDPCE and two underlying IP domains, namely D_1 and D_2, managed by two corresponding HDPCEs. First, we set up lambda paths that have ten units of bandwidth from each BR in D_1 to each BR in D_2. Therefore, after the lambda path setups, there are nine lambda paths in the network, as shown in Fig. 7(a). After setting up lambda paths in the GMPLS networks, we set up IP paths that require one unit of bandwidth from the underlying inner router, R1, to the other inner router, R2, using two different backup methods.

In this evaluation, β in Eq. 2 is set to 10. As shown in the figure, preplanned restoration backup accommodates 20 – 80% more primary IP paths than protection backup at the expense of vulnerability against simultaneous malfunctions. The larger the value of α is, the lower the accommodation rate of primary IP paths is in the protection method, because this network, which has only 9 lambda paths, is a relatively small network. Preplanned restoration, however, has a consistent accommodation rate regardless of the value of α; thus, the network operator of a VPN

provider does not have to adjust the value of α. The active restoration method accommodates the most primary paths among the three methods, though the SRLG-disjointed backup path is not necessarily assured.

5 Conclusion

We described an HDPCE-based VPN service that provides IP domain customers with end-to-end QoS assurance, security, and three flexibly chosen backup methods.

First, HDPCEs provide individual-flow-based multidomain routing, which assures the required bandwidth of each flow. Second, link-state information of each customer is never revealed to other customers, so security is assured. Third, three backup methods: protection, preplanned restoration, and active restoration are flexibly chosen depending on the priority and backup strategy of each path of the customer.

References

1. A. Bonerjee, J. Drake, J. P. Lang, and B. Turner, "Generalized multiprotocol label switching: an overview of routing and management enhancements," IEEE Commun. Mag., vol. 39, no. 1, pp. 144-150, Jan. 2001.
2. H. Matsuura, T. Murakami, and K. Takami, "An Interdomain Path Computation Server for GMPLS Networks," IEICE Trans. Commun., Vol. E88-B, No. 8, pp. 3329-3342, August 2005.
3. H. Matsuura, N. Morita, T. Murakami, and K. Takami, "Hierarchically Distributed PCE for GMPLS Multilayered Networks," IEEE Globecom 2005, St. Louis, Missouri, USA, Nov. 2005.
4. R. Bless, "Towards Scalable Management of QoS-based End-to-End Services," IEEE/IFIP Network Operations and Management Symposium, April 2004.
5. Performance Evaluation for a DiffServ Network's PHBs EF, AF, BE and Scavenger," IFIP/IEEE International Symposium on Integrated Network Management, May 2005.
6. Sun Microsystems, "Enterprise JavaBeans™ Specification Version 2.1," August 2002.
7. J. P. Vasseur, J. L. Roux, A. Ayyangar, E. Oki, A. Atlas, and A. Dolganow, "Path Computation Element (PCE) communication Protocol (PCEP) Version 1," IETF Internet Draft, draft-vasseur-pce-pcep-02.txt, Sep. 2005.
8. Object Management Group, "CORBA V2.3.1," Oct. 1999.
9. N. Bitar, R. Zhang, and K. Kumaki, "Intar-AS PCE Requirements," IETF Internet Draft, draft-bitar-zhang-inter-AS-PCE-req-011.txt, Oct. 2005.
10. J. L. Roux, "PCE Communication Protocol (PCECP) specific requirements for Inter-Area (G)MPLS Traffic Engineering," draft-leroux-pce-pcep-interarea-reqs-00.txt, Oct. 2005.
11. E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathemakik, 1, pp. 299-271, 1959.
12. Q. Ma and P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees," In Proceedings of IEEE International Conference on Network Protocols, October 1997.
13. D. Xu, Y. Xiong, C.Qiao, and G. Li, "Trap avoidance and protection schemes in networks with shared risk link groups," in IEEE Journal of Lightwave Technology, Special issue on Optical Network, Nov. 2003.
14. E. Oki et al., "A Disjoint Path Selection Scheme with SRLG in GMPLS networks," Proc. of IEEE HPSR'2002, 88-92, May 2002.

WBEM-Based SLA Management Across Multi-domain Networks for QoS-Guaranteed DiffServ-over-MPLS Provisioning*

Jong-Cheol Seo[1], Hyung-Soo Kim[2], Dong-Sik Yun[2], and Young-Tak Kim[1,**]

[1] Dept. of Information and Communication Engineering,
Graduate School, Yeungnam University
214-1, Dae-Dong, Kyungsan-Si, Kyungbook, 712-749, Korea
[2] Network Technology Laboratory, System Architecture Research Division,
Solution Research Department, Korea Telecom (KT)
463-1, Jeonmin-dong, Yuseong-gu, Daejeon, 305-811, Korea
sjc2305@paran.com, essence@kt.co.kr,
dsyun@kt.co.kr, ytkim@yu.ac.kr

Abstract. For QoS guaranteed DiffServ-over-MPLS connection provisioning in multi-domain networks, the managed objects in each domain should be globally standardized and publicly accessible by other NMSs. And each NMS participating in the inter-domain networking should negotiate with other NMSs for end-to-end service provisioning. In this paper, we propose SLA negotiation by COPS protocol and design the managed objects by expanding existing experimental DMTF CIM MOFs. We propose service location protocol based directory agent to register/discover the NMSs' that participate in the multi-domain TE-LSP connection provisioning. For better performance, we design the multi-threaded provider implementations on OpenPegasus WBEM platform. The detailed implementations of the providers with performance analysis are explained.

1 Introduction

In order to provide on demand QoS-guaranteed DiffServ-over-MPLS service provisioning across multiple autonomous system (AS) domain networks, the network management system (NMS) should be able to collect the information of inter-AS connectivity and available network resources among autonomous system boundary router (ASBR) ports, and fault restoration capability of the traffic engineering label switched path (TE-LSP) [1]. For this end-to-end QoS-guaranteed DiffServ provisioning management, NMS must establish service level agreement (SLA) with other providers and with the customers. The NMS needs to interact with other NMSs of different network providers for the QoS-guaranteed TE-LSP connection establishment and this interaction is required to be platform and network resource

* This research was supported by the MIC, under the ITRC support program supervised by the IITA.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 312 – 321, 2006.
© Springer-Verlag Berlin Heidelberg 2006

independent. The managed objects (MOs) in inter-AS traffic engineering should be standardized in public domain, thus can be easily accessible by other network providers. Currently, network management MOs for QoS-guaranteed DiffServ connection provisioning is not well standardized in public domain. Most currently available commercial IP/MPLS Systems are supporting simple network management protocol (SNMP) or command line interface (CLI) based management function without globally standardized MO definitions. As a result, it is not possible to easily collect the networking resource information of inter-AS transit networks from other NMSs, negotiate the possible differentiated service provisioning on SLA negotiation, and establish and manage flexible TE-LSPs across multiple domain networks.

Since web-based enterprise management (WBEM) and common information model (CIM) are platform and resource independent distributed management task force (DMTF) standard, they can provide a good framework for design and implementation of the MOs for inter-AS traffic engineering [1]. WBEM defines both a common model (i.e., description) and protocol (i.e., interface) for monitoring and controlling resources from diverse sources (e.g., different types of platforms or different types of resources on the same platform). The inheritance and object-oriented design from CIM MOF (managed object format) can reduce development time, and also provide independence in platform and programming language [2].

The DMTF CIM MOFs are used as base classes for inter-operability among management systems that would be implemented by different network operators. The MOF design with inheritance enables simple implementation of new MOs for new management function with reusable common information of base MO, as the object-oriented design approach provides. Also, the polymorphism for easy networking enables easy access to new MO by remote manager who has limited knowledge on the actual implementation of inter-AS traffic engineering MIB (management information base). Currently CIM has some experimental MPLS (multi-protocol label switching) and DiffServ (differentiated service) related MOFs, but detailed MOFs for QoS-guaranteed DiffServ provisioning are not available.

We have designed MOs for SLA management, and inter-AS TE-LSP connection management with several extensions on existing experimental DMTF CIM MOFs with inheritance. With WBEM architecture, we have designed and implemented SLA negotiation. We have used directory agent (DA) [3], a centralized repository for storing the details of NMSs and their corresponding service location information. In this paper, we explain the detailed implementation approaches of WBEM-based TE-LSP connection management and SLA negotiation across multiple domain networks using OpenPegasus [4], OpenSLP [3], and COPS (common open policy service) [5].

The rest of this paper is organized as follows. In section II, the related work on WBEM and service level subscriptions (SLS) management are briefly introduced. In section III, WBEM based architecture for TE-LSP connection establishment and SLA negotiation are explained. The DMTF CIM-based extended MOF design for SLS negotiation, connection establishment and necessary WBEM providers are explained in section IV. In section V, the performance analysis of WBEM-based architecture for connection establishment is explained. We finally conclude this paper in section VI.

2 Background

2.1 Web-Based Enterprise Management (WBEM)

WBEM is a platform and resource independent DMTF standard that defines both a common model and protocol for monitoring and controlling resources from diverse sources. DMTF defines the CIM that defines the management information for system, networks, application and services, etc. CIM includes the description of a meta-language for describing data (i.e., CIM specification) and a description of the resources to be managed (i.e., CIM schema and vendor extensions). CIM-XML is a DMTF standard for WBEM communication protocol that uses the CIM specification and CIM schema for the representation of managed resources, defines the encoding of CIM data and operations into XML, and uses HTTP transport protocol for exchanging CIM-XML encoded requests and responses [2]. CIM MOF is the language defined by the DMTF for describing classes and instances.

2.2 Service Level Specification (SLS) Management

The SLS is used to denote the technical characteristics of the service offered in the context of an SLA. The service technical characteristics refer to the provisioning aspects of the service, such as request, activation and delivery aspects from network perspectives. Non-technical service provisioning aspects such as billing and payment aspects, are not part of the SLS; they are part of the overall SLA. SLS forms the basis of the agreements between providers for traffic exchange in the Internet. SLS include SLS identification, scope (geographical/topological region), flow identification, traffic conformance, excess treatment, performance guarantees, schedule, etc. Our work draws from the SLS template specification work of the MESCAL SLS specification [6 - 8].

QoS-based services are offered on the basis of the SLAs, which set the terms and conditions on behalf of both providers and customers in providing and requested services, respectively. Before committing any connection establishment, SLA negotiation should have been made between the providers. There is a performance monitoring part, for each negotiated QoS parameters (such as delay, jitter, packet loss, and throughput), which is composed of (i) Measurement Period, (ii) Reporting, and (ii) Notification Threshold [7]. The performance management would perform monitoring the QoS parameters that had been negotiated in the SLA so as to check that whether SLA is violated or not.

3 Architecture of WBEM-Based SLA Management on Multi-domain Networks

3.1 WBEM Based SLA Management Functional Architecture

Fig.1 depicts the functional block of WBEM-based SLA management for inter-AS TE-LSP connection establishment. The OpenSLP server acts as a dedicated DA where all NMSs register and discover the service contact points using service location protocol (SLP).

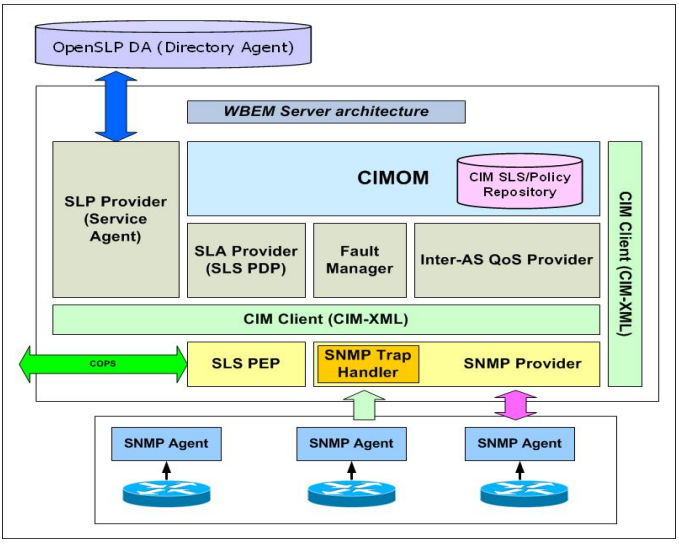


Fig. 1. WBEM-based SLA Negotiation Functional Block

Each NMS is configured with OpenPegasus WBEM Server. OpenPegasus 2.5.1 supports service agent (SA) functionality for service registration to DA. The API (application programming interface) for OpenPegasus client includes user agent (UA) functionality such as service lookup and request. The SLP provider has SA functionality for service registering and CIM client has UA functionality for service request. During the NMS's WBEM initialization, the SA in SLP provider tries to detect DA and then it registers itself to the DA. The service type has been defined as "service:wbem.InterAS:pegasus". Once the service registration is successful, the UA can directly request service discovery to DA, and get information about interested service contact points.

For SLA negotiation, we have designed the SLA Provider with COPS PDP and PEP functionalities. The Inter-ASQoSProvider is designed and implemented for the purpose of inter-AS Traffic Engineering for QoS-guaranteed DiffServ-over-MPLS provisioning. SNMP Provider can interact with SNMP supported network element (NE) directly, using SNMP protocol. SNMP Trap Handler is included in SNMP Provider, and is used for fault detection mechanism by Fault Manager.

3.2 Interaction Among NMSs for SLA Management

The interaction among the NMSs, DA and CNM (customer network manager) for SLA negotiation is shown in the Fig. 2 Before the interaction among NMSs is done, the NMS need to perform initialization of its providers. During NMS provider initialization, it tries to register itself to the DA, and to discover other NMSs that participate in inter-AS TE from the DA using SLP protocol. Each NMS tries to gather the resource details from other NMSs to make inter-AS topology configuration. After the topology configuration is accomplished, SLA negotiation for inter-AS TE can be handled.

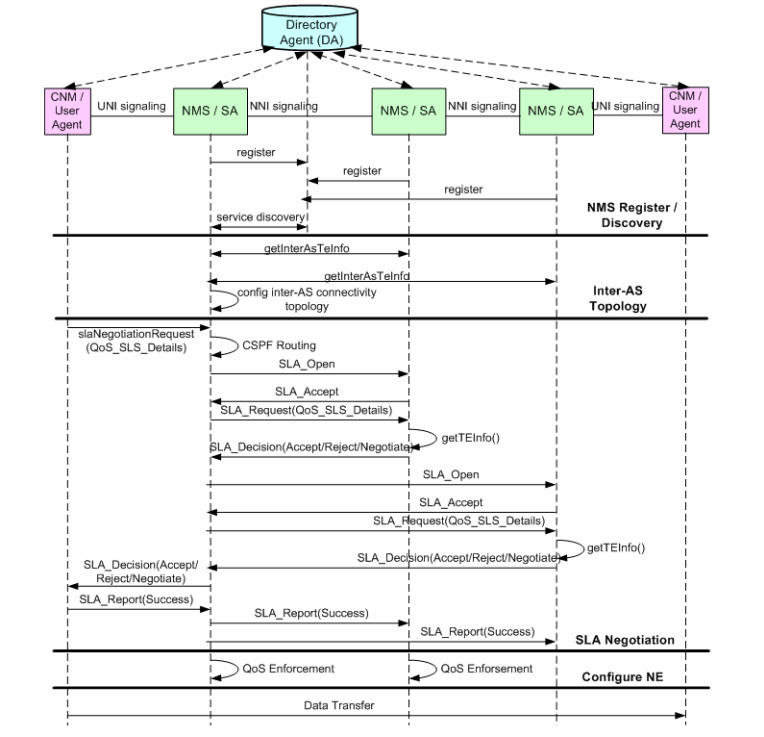


Fig. 2. Interaction sequence diagram for SLA negotiation in inter domain networks

When on demand QoS-guaranteed TE-LSP connection request is made from any CNM, the ingress NMS performs constraint-based shortest path first (CSPF) routing. Then it makes SLA negotiation among the providers which manages the corresponding domains that are in the path to the destination. Once the ingress NMS receives successful decision from other providers for the negotiated parameters, it can make QoS enforcement in its domain, i.e., it can configure its managed resources for accommodating the traffic agreed on the negotiated SLA. All the service providers who agreed with the SLA will perform QoS enforcement on their managed domains. The data transfer can take place at the scheduled time mentioned as one of the SLA parameters during SLA negotiation.

4 Design and Implementation of CIM MOF-Based MOs and Providers for WBEM-Based SLA Management

4.1 Extended CIM MOs for SLA Management

Fig.3 depicts unified modeling language (UML) representation of CIM classes and associations designed for SLA negotiation and connection establishment in inter-AS TE. Some classes are designed as abstract classes and other classes inherit from the

super classes. The classes QoS_interASNet_SLASrv, QoS_InterDomain_SLS, and Qos_DiffServOver MPLS-InterDomain_SLS are designed for SLA negotiation between the NMSs. The SLS parameters such as scope, flow-id, traffic conformance, excess treatment (drop/remark/shape), performance guarantees (delay, jitter, loss, throughput), schedule (time, day, year etc.) and performance monitoring parameters are defined in the base class QoS_InterDomain_SLS. The class Qos_DiffServOverMPLS-Inter-Domain_SLS extends from QoS_InterDomain_SLS with TE parameters.

For inter-AS traffic engineering, we designed QoS_OAM_Service MO which acts as super class and the services such as QoS_interASNet_PMSrv, QoS_interASNet_FMSrv, QoS_interASNet_CMSrv for performance management, fault management and connection management respectively. The class QoS_interASNet_SLASrv has the SLA negotiation module with both customer and provider implementation.

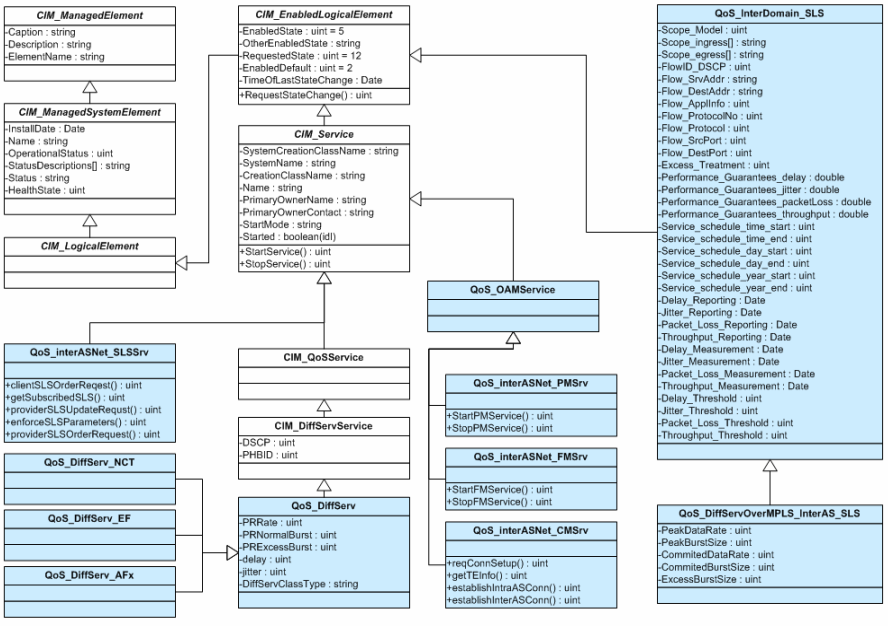


Fig. 3. DMTF CIM based MOF classes for SLA negotiation

4.2 Inter-AS QoS Provider Implementation

The inter-AS QoS provider which is shown in Fig.1, acts as core part of our WBEM architecture. This provider includes instance provider, association provider and method provider functions for inter-AS TE (traffic engineering). When WBEM server is loaded, the Inter-AS QoS Provider gets service location list of other providers from SLP provider and configures the inter-AS connectivity topology. When SLA is negotiated by the customer (or other providers) with requested parameters, it

computes CSPF routing for the requested destination based on the up-to-date inter-AS topology details. When NMS needs to establish TE-LSP connection setup and backup path, the method provider starts to establish connection by interacting with SNMP provider or CLI Provider in the WBEM server. All the provider modules are implemented using C++.

4.3 SLP Provider Implementation

SLP provider in NMS consists of a service agent, user agent and CIM client. When the WBEM server is loaded, the SLP provider registers Inter-AS supported WBEM Pegasus service to statically registered DA. SLP provider periodically sends unicast service request message to DA to get the service location list of registered NMSs. Using CIM SLP Template instances, the SLP provider stores the details of the registered NMSs.

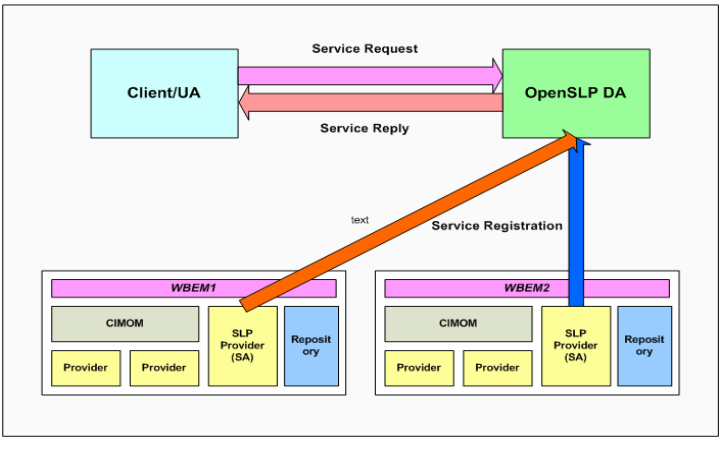


Fig. 4. Service Discovery by SLP Provider

Fig.4 shows the SLP provider operation for service register and discovery. In our implementation, DA address is statically configured in the NMS's SLP provider. The DA is implemented in Linux platform using OpenSLP[3]. The UA functionalities are provided by the OpenPegasus 2.5.1 client API.

4.4 SLA Provider Implementation

SLA provider is used to negotiate the SLA between the providers and the customers. Before establishing any data transfer for on-demand connection request, the providers and customers negotiate each other for the mutual agreement on the proposed parameters in the SLS. The SLA negotiation between the providers and customers are implemented using the WBEM CIM model. Fig.2 explains the SLA negotiation process between the customers and the providers. The client initiates the negotiation by SLA_Open() to the provider. The SLA provider accepts the negotiation session by

SLA_Accept(). When the client sends the SLA_Request() with the required parameters for on-demand TE-LSP connection provisioning, the SLP provider checks whether the managed domain has enough resource for the new connection. This is done by calling getTEInfo(), which in-turn calls getUptodateLocalStateInfo() method of Inter-AS QoS provider. The current network resources are checked and if the managed domain has enough resource, the service level negotiator will initiate the service level negotiations to other SLP providers who are in the route to the destination. If there is not sufficient resources available, the SLP provider can re-negotiate to the requested parties by calling SLA_Decision() request with re-negotiation option. Once the ingress NMS receives successful SLA_Decision() from all the NMSs, they can enforce the NE configuration by using SNMP provider or CLI provider. The ingress NMS calls the customer with the decision by SLA_Desicion(). The decision could be either accept/reject or re-negotiate. The customer can send the report by SLA_Report() to the ingress-NMS, which conveys that the customers decision to the provider about the negotiation.

5 Performance Evaluations on SLA Negotiation for TE-LSP Connection Establishment in Inter-domain Networks

In our current implementation with OpenPegasus based WBEM architecture for SLA negotiation in 5 AS domain networks (as shown in Fig. 2), the overall negotiation time among the providers takes around 6~8 seconds. On the average it takes around 1.8 sec for successful provider-to-provider negotiation.

5.1 SLA Negotiation for TE-LSP Connection Establishment

Fig. 5 shows time taken by the SLA negotiation process with five NMS. When the ingress NMS gets a request for SLA negotiation for TE-LSP connection establishment, it checks itself with getTEInfo(). The request will perform association traversals and instance enumerations on the repository. The graph depicts the time taken for enumerating the instances the CIMOM repository.

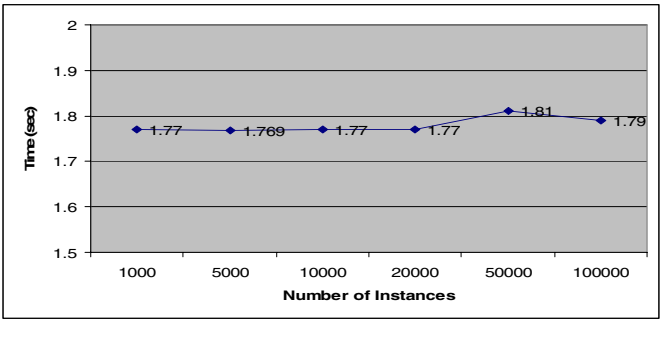


Fig. 5. SLA negotiation request processing

From the graph, it is clear that there is not much variance in time taken and the time taken is irrespective to the number of instances enumerated. On the average, it takes almost 1.78 seconds. It shows the better scalability for the number of instances to be enumerated. The ingress SLA provider will interact with other four NMSs SLA providers sequentially and the cumulative time taken is around 6~8 sec for end-to-end negotiation.

5.2 Performance Measurements for All Activities in End-to-End SLA Negotiation

Fig. 6 shows the complete time taken by each functions of WBEM architecture involved for SLA negotiation for TE-LSP connection establishment in inter-domain networks. The service discovery using SLP protocol took around 7 ms and the CSPF computation for the given topology with five NMS took around 0.252. In general the method invocation such as SLA_Open(), SLA_Report(), and SLA_Decision() take few milliseconds (30~40 ms).

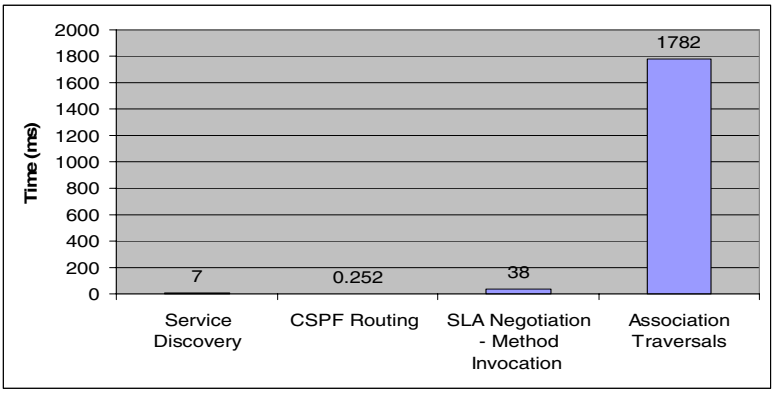


Fig. 6. Complete SLA negotiation activities

From the performance analysis, we found that the instance creation time depends on the hardware specification and method invocation does not take much time. The Provider API function for association traversals and enumerating instances also does not take much time. We are developing multi-thread based parallelism in implementation to improve the overall performance.

4 Conclusions

In this paper, we designed MOs for SLA negotiation for inter-AS TE-LSP connection establishment with extensions on existing experimental DMTF CIM MOFs with hierarchical inheritance. We designed MOs to represent SLA negotiations, service registration/discovery, interASNet & QoSDiffServNet. We also designed MOs related

to interASNet OAM functions, such as connection management, performance monitoring and fault management.

The interaction scenario among DA, NMSs with WBEM server/client function of different AS domain networks for SLA negotiations and inter-AS TE are explained. The SLA negotiation is done by using COPS protocol. Currently we are implementing the providers illustrated in the proposed inter-AS traffic engineering system based on OpenPegasus WBEM. The implementation is done on Linux based OpenPegasus WBEM. From the performance analysis, the overall time taken for SLA negotiation for TE-LSP connection establishment between the provider and the customer is around 1.7~2 seconds. Currently we are doing the detailed performance analysis. From the result of the analysis and the standardized WBEM based architecture, we can conclude that WBEM-based SLA negotiation architecture for inter-AS traffic engineering can be successfully applied for inter-AS QoS-guaranteed DiffServ-over-MPLS connection provisioning.

References

1. Young-Tak Kim, “Inter-AS Session & Connection Management for QoS-guaranteed DiffServ Provisioning,” Proc. of International Conference on Software Engineering Research, Management & Applications (SERA 2005), Mt. Pleasant, USA, pp. 325~330.
2. Web-based Enterprise Management (WBEM),
3. http://www.dmtf.org/standards/wbem/.
4. OpenSLP, www.openslp.org.
5. OpenPegasus, http://www.openpegasus.org/.
6. T.M.T. Nguyen, N. Boukhatem, G. Pujolle, "COPS-SLS Usage for Dynamic Policy-based QoS Management over Heterogeneous IP Networks", IEEE Network, May/June 2003.
7. SLS Management, http://www.mescal.org/deliverables/d1.3_finalv2.pdf.
8. Goderis, D. et al., “Service Level Specification Semantics and Parameters”, Internet Draft, <draft-tequila-sls-02.txt>, January 2002.
9. Dong-Jin Shin, Young-Tak Kim "Design and Implementation of Performance Management for the DiffServ-aware-MPLS Network," Proceedings of Conference on APNOMS 2003, Fukuoka, Japan, October 2003.
10. Shanmugham Sundaram, Abdurakhmon Abdurakhmanov, Young-Tak Kim, "WBEM-based Inter-AS Traffic Engineering for QoS-guaranteed DiffServ Provisioning," IEEE Broadband Convergence Networks (BcN2006) Workshop, Vancouver, British Columbia, Canada, 2006.

Network Support for TCP Version Migration

Shingo Ata, Koichi Nagai, and Ikuo Oka

Graduate School of Engineering, Osaka City University, 3–3–138 Sugimoto, Sumiyoshi-ku, Osaka 558–8585, Japan

Abstract. Recently, new versions of the TCP protocols such as HSTCP (High Speed TCP) has proposed to adopt the rapid growth of the Internet. However, it is important to consider a deployment scenario to shift gradually from the current to the new versions of TCP. A fairness problem between the performance of both current and new versions of TCP will be arisen when they share the same bottleneck link, which leads some complaints from people using the current version of TCP. In this paper, we first clarify the relation between the deployment share of HSTCP and the negative impact to TCP Reno users. We then propose a new management mechanism which supports TCP version migration from TCP Reno to HSTCP based on the relation. Through simulation, we also show that our mechanism enables to accelerate a gradual migration of TCP versions by achieving the end users' complaints as little as possible.

1 Introduction

In recent years, speedup of backbone line progresses in the Internet, and, in congestion controls of TCP (Transmission Control Protocol) used in the Internet mainly, it is considered a technique to improve to be able to follow a broadband of backbone line.

Currently a version called *Reno* is widely used for TCP about a congestion control in the Internet. However, with a broadband of the Internet, it becomes clear that we cannot deal with speedup of a network only by just using congestion control algorithm of TCP Reno. Therefore there are number of researches about new version of TCP to make improvement to TCP Reno. As such a new TCP, HighSpeed TCP (HSTCP) [1]Fast TCP [2]Scalable TCP [3] are proposed. HSTCP, for example, does increase spreading of a congestion window size greatly for every one RTT in comparison with the one of Reno, and decrease small of window size when a congestion occurred. It thus keeps a large quantity of window sizes during a communication. As a result, a HSTCP can support a line speed of Gbps order.

However, because TCP is a congestion control protocol performed between transmit and receive end nodes, it is necessary to replace the TCP protocol stack running on the transmit or the receive (or sometimes both) node in order to use a new version of TCP. As a result, more than one versions of TCPs coexist in the same network line in the migration stage. At this time problem of fairness

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 322–331, 2006.
© Springer-Verlag Berlin Heidelberg 2006

about communication performance occurs between the users who use different version of TCP, because difference in performance occurs between both versions even under a similar network environment [4,5].

There are following two important issues to complete a step by step migration between existing and new TCP versions.

1. *A new version is to be good in performance compared to the existing version:* It is an end user or a service-provider (and not a network provider) that judges whether it replaces TCP stack to a new version. There is no incentive for users to shift a new version of TCP unless explicit effect (e.g., gain additional throughput, or speed does not fall even if packet loss occurs) is shown.
2. *Not bringing remarkable performance degradation for a user using the existing version:* A user (or service provider) cannot but become careful for introduction of a new version from a viewpoint of stability use of network, if a congestion control of a new version gives the one of the existing version big effect, and, as a result, remarkable deterioration produces it in performance. It has been considered much TCP versions by many researchers, but TCP Reno is still used as the mainstream till now. It is because the effect that would be occurred is not investigated enough in the environment where the new version is in conjunction with TCP Reno. Currently there are enormous nodes connected to the Internet, and it also indicates that the instability of the control of TCP Reno directly means the instability of the Internet. From the viewpoint of network operation, it is undesirable to do a version change only by the reason of a respect to be efficient in performance.

These have a relationship to disagree each other. In other words, an incentive to introduce a new version would be increasing according to its performance advantage, while it would also be careful from the viewpoint of stability use of a network. On the other hand, it can drive forward with safe when the compatibility between a new and the existing TCP versions is high, but if there is no remarkable improvement in performance, merit of migration is not felt for a user.

We propose a new model for TCP version migration by the network support in this paper. We consider how a network provider (ISP) should promote a migration to a new version of TCP with minimizing complaints from subscribers who use the existing version. The advantage on a step-by-step migration of TCP version by support of the network is that a new congestion control can be designed more suitable without being conscious of an upper compatibility of the existing version. It is possible that the network provider promotes the advantage of a new version widely while minimizing complaints from existing version users.

We consider a step-by-step migration model of TCP version as shown in Fig. 1. In this figure, the horizontal axis shows a time progress, and the vertical axis shows the deployment share of each version. We can divide migration roughly into following four phases.

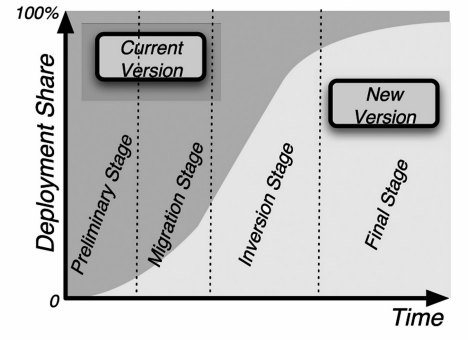


Fig. 1. Strategy on TCP version migration

1. *Preliminary Stage:* There are extremely a few users deploy a new version experimentally. Because it is important that in this phase the advantage of a new version should be recognized widely, we should not do any constraints for a new version.
2. *Migration Stage:* The validity of a new version is recognized by some users, and the number of users who updated to a new version is slowly. Fairness with the existing version users becomes the problem in this phase. Therefore, it becomes important to control complaints by the existing version users as minimum as possible to recognize validity of a new version more widely.
3. *Inversion Stage:* When the merit of a new version is recognized widely, many users are starting a change to a new version. It is expected that the number of the updating users suddenly increases in this phase. It is because a new version is implemented into standard function (APIs) of operating systems (OS), and users can make a change to a new version by only invoking an update of the operating system.
4. *Final Stage:* Most users have completed the change to a new version. We can complete the change by recommending transference (in other words, update of OS) to a new version for the existing version user individually.

Among the above-mentioned stage, it is *migration stage* that ISP has to support mainly. It becomes important that we show an advantage in performance of a new version while minimizing complaints of the existing version users. Therefore, in this paper, we introduce subjective value (MOS:Mean Opinion Score) for a file download as a metric to show the degree of complaints from the existing version users. We then propose a new network support model to realize performance advantages of a new version while achieving the MOS value where the existing version users do not complaint. In this paper, we consider about step-by-step migration from TCP Reno to HSTCP.

This paper is organized as follows. We first describe the impact HSTCP on the migration stage in Section 2. We next model the relation between the performance degradation for a file download and subjective scores in Section 3. Based on these results, we propose a network support model to promote migration from

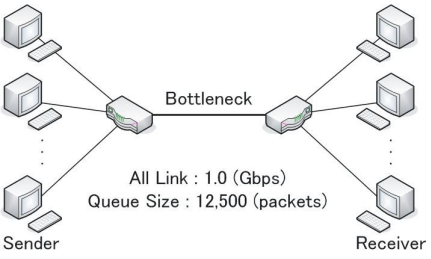


Fig. 2. Simulation environment

TCP Reno to HSTCP while maintaining subjective value in Section 4. Finally, we describe a summary with future topics in Section 5.

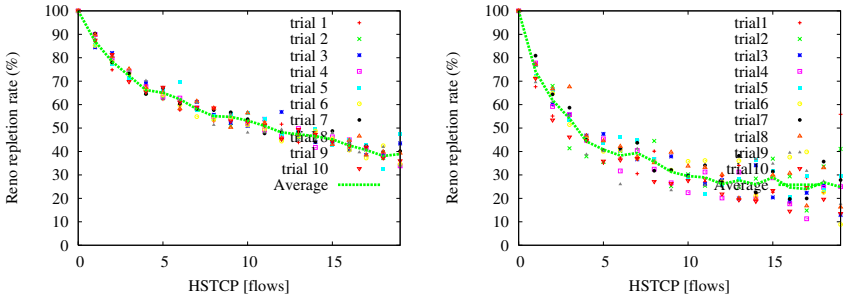
2 Influence of HSTCP to TCP Reno

In this section, we show some simulation results to investigate the impact of HSTCP to TCP Reno flows.

We use uses ns-2 (ns-2.1b9) for simulation. The network topology we used is shown in Fig. 2. All nodes between the sender and the receiver sides share a single (Bottleneck) link, whose capacity is 1 Gbps and transmission delay is set to be 50 msec. We also set the size of router buffer to 12,500 packets. We use two principles (RED; Random Early Detection and Drop-tail) as queue management algorithm in the router. We enable a SACK (Selective ACK) option in both TCP Reno and HSTCP. Due to the space limitation we omitted results without SACK option because simulation results are the same. We use FTP (File Transfer Protocol) as application. Furthermore, we setup small TCP flows and a set of Web flows as back traffic.

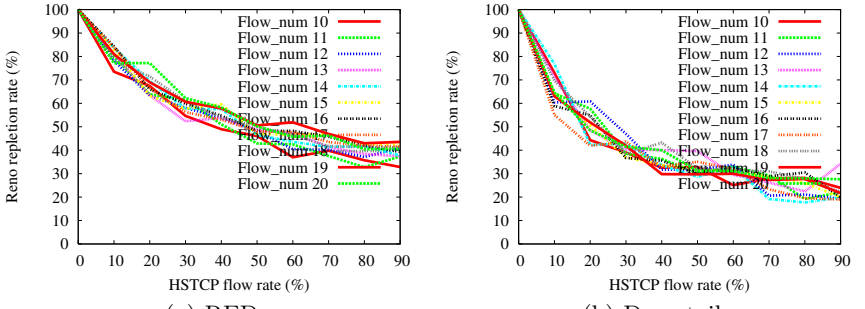
We first investigate the degree of throughput degradation by increasing the number of HSTCP flows. We set total number of sender-receiver pairs to be 20, i.e., total 20 flows share the bottleneck link in Fig. 2. First, we set all flows to TCP Reno, and obtain the throughput of each flow. We then change the version of TCP for one flow to HSTCP, and obtain the throughput. We define the ratio of the throughput when HSTCP and TCP Reno flows are mixed to the one when all flows are TCP Reno as *satisfaction ratio*. For example, the average of throughput of TCP Reno when one flow is changed to HSTCP is 8 Mbps, and the average throughput when all flows are TCP Reno is 10 Mbps, the satisfaction ratio is 80%.

By increasing the number of flows changed to HSTCP from 1 to 20, we evaluate the satisfaction ratio for TCP Reno flows. Figs. 3(a) and 3(b) show results of satisfaction ratio according to the number of HSTCP flows with RED and Drop-tail routers respectively. From these figures, we can observe that the satisfaction ratio is decreasing when the number of HSTCP flows increases. Moreover, we cannot observe any significant difference on the degradation of the satisfaction



(a) RED (b) Drop-tail

Fig. 3. Satisfaction ratio according to number of HSTCP flows (20 flows)



(a) RED (b) Drop-tail

Fig. 4. Satisfaction ratio according to ratio of HSTCP flows

ratio when the number of HSTCP flows is 15 or above. By comparing these figures, a Drop-tail router makes more significant impact to TCP Reno flows than a RED router.

We next investigate the impact of the ratio of HSTCP flows to all flows with changing total number of all flows from 10 to 20. The result is shown in Fig. 4(a) and 4(b), where the satisfaction ratio dependent on the ratio of HSTCP flows are plotted in both RED and Drop-tail cases. The same tendency can be found from these figure, however, one additional observation is that the impact to TCP Reno is not characterized by the number of HSTCP flows, but by the ratio of HSTCP flows in all flows.

3 Modeling Users' Satisfaction with Subjective Score

In this section we clarify how much a user feels complaint due to transmission rate of TCP Reno limited by HSTCP with subjective score.

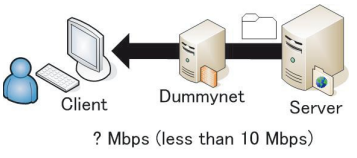


Fig. 5. Experimental environment

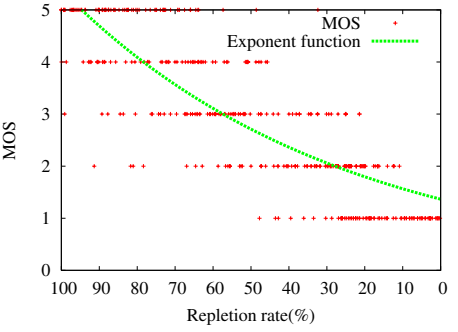


Fig. 6. Relation between MOS values and satisfaction ratios

Table 1. Mean Opinion Scores

MOS value	Validation criterion	Satisfied?
5	Almost the same as base rate	Yes
4	Feel delays but negligible	Yes
3	Sometimes feel delays but marginal	Yes
2	Sometimes feel unacceptable delays	**No**
1	Almost delayed or feel significant delays	**No**

3.1 Experiment Summary

For subjective evaluation experiment we use a MOS (Mean Opinion Score) value that is a mean value of subjective satisfactory of users. The objective of the experiment is to model a relation between the performance degradation of TCP Reno flows and the degree of end users' complaints.

In this paper we focus on the difference subjective score of users by changing the speed of transmission in data download application.

We configure a dymmynet running on FreeBSD [6] between server and client in order to emulate the bandwidth of the link between the server and the client. For each evaluator, we proceed following steps to obtain the subjective score.

1. Set the bandwidth to 10 Mbps (we call it as *base transmission rate*) and invoke a file download of 10 MB by FTP to understand the speed of 10 Mbps.
2. Set the bandwidth on the link with a random value from 0.1 Mbps to 10 Mbps by changing the configuration on Dummynet (we refer as *examined transmission rate*).
3. Invoke the 10 MB file download again and ask the score (one to five) shown in Table 1 compared with the speed of 10 Mbps.

We repeat fifteen tests for each evaluator, i.e., totally 165 samples are obtained.

3.2 Impact of Subjective Scores Against Satisfaction Ratio

We show the relation between subjective scores and throughput degradations in Fig. 6. We use the normalized ratio (called *degradation ratio*) of the examined transmission rate to the base transmission rate (10 Mbps) instead of the direct value of the examined transmission rate, so that we directly map this ratio to the satisfaction ratio. In this figure, the horizontal percentage is the degradation ratio of the examined transmission rate.

As observed in Fig. 6, most of people do not feel complaint when the degradation ratio is more than 80%, however, the MOS value decreases significantly at the region where the degradation ratio decreases 80% to 40%, and the MOS value becomes under 3 when people receive 50% of performance degradation compared with their usual cases. When the degradation ratio is around 30%, most of people feel bad on their performance, and nobody feels satisfied when the ratio becomes 10% of the original performance. To model the relation between the degradation ratio and the MOS value we apply an exponential function by using a least square method, which is the same approach shown in [7]. The result of applying an exponential function is shown by the curve in Fig. 6. As the result of fitting, we give the subjective score (MOS) $S(R_d)$ for the degradation ratio R_d by

$$S(R_d) = 1.416 \times e^{0.0138 \times R_d}. \quad (1)$$

3.3 Impact of HSTCP Flows to Subjective Scores

By using both two relations (i.e., relation between the satisfaction ratio and the ratio of HSTCP flows, and relation between the degradation (satisfaction) ratio and the subjective score), we finally obtain the relation how the ratio of HSTCP flows affects the MOS value. For simplicity we directly map the degradation ratio to the satisfaction ratio. We also use the average value of satisfaction ratios which are obtained by changing the total number of flows from 10 to 20, because as previously described the satisfaction ratio is mainly affected by the ratio of HSTCP flows (i.e., not by the number of HSTCP flows). By applying Eq. (1) we derive the relation between the ratio of HSTCP flows and MOS values shown in Fig. 7.

In this paper, we suppose that users feel complaint when the MOS value becomes less than three. Under such definition we can observe that end users' complaint occurs when the ratio of HSTCP flows becomes around 40% to 50% of all flows in RED router case. However, the MOS value still remains about 2.4 even if the ratio of HSTCP flows becomes 90% of all flows. In other words, RED routers are more suitable than Drop-tail routers when we consider the migration of TCP versions. On the other hand, with Drop-tail routers, the MOS value becomes less than 3 though the ratio of HSTCP flows is a few (i.e., 10% to 20%). Moreover when the ratio of HSTCP flows reaches to 80% the MOS value becomes less than 2, where most people have strong complaints on their performance. From these results we propose a method to achieve the MOS value for TCP Reno users more than 3 by the network support in the following section.

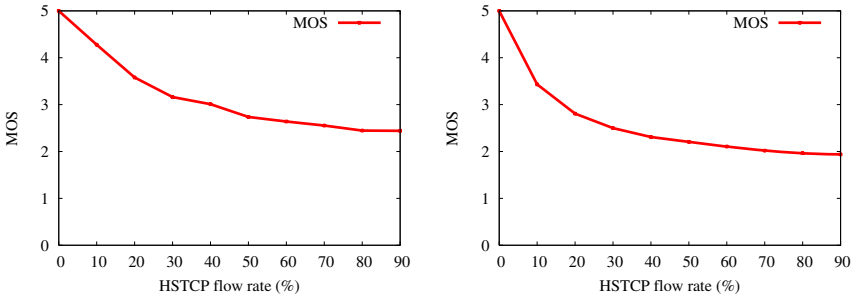


Fig. 7. MOS values against ratio of HSTCP flows

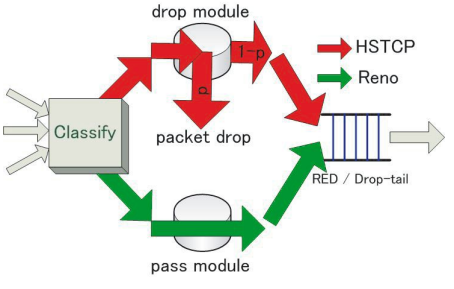


Fig. 8. Router structure for network support

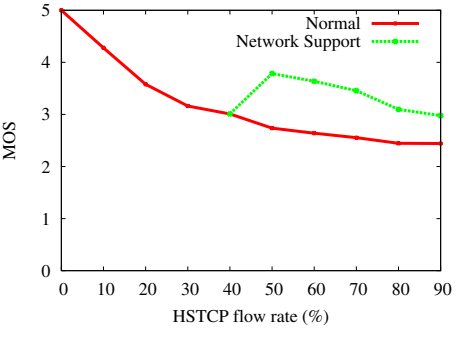


Fig. 9. Result of network support (RED)

4 Migration Scheme by Network Support

4.1 Architectural Overview

We show the mechanism on router to realize the network support to maintain the MOS value for TCP Reno users in Fig. 8. We perform our network support at the edge router of the ISP. We focus on egress flows (i.e., flows outbound to the Internet) as the target traffic for the network support, because the network support should be only performed at the first edge of the network.

For each packet arrival, the router classifies the packet into HSTCP or TCP Reno flows. Note here that it requires an identification method for HSTCP traffic in addition to a generic flow classification mechanism. However in this paper we do not consider the detail of identification method, which is one of our future research topics. One promising approach is to identify HSTCP flow by comparing the growth of its rate for each round trip time.

If the packet is identified as HSTCP flow, the packet is forward to the *drop module*, otherwise it is forwarded to the *pass module*. The drop module drops the

forwarded packet with the random probability p_d, i.e., the packet is forwarded to the buffer of the router with the probability $1 - p_d$. The pass module is a single packet buffer and simply passes the packet to the router buffer. The purpose of the pass module is to keep the order of packet arrivals. After stored into the buffer, the packet is forwarded to the next hop by the normal operation on the router.

4.2 Determination of Packet Dropping Rate

The key of the network support is how to set the dropping probability p_d. Based on the results shown in Fig. 4, we first model the relation between the ratio of HSTCP flows and the satisfaction ratio of TCP Reno flows by using an exponential function. With the least square method, the satisfaction ratio on RED and Drop-tail are given by $f_{RED}(H) = 91.41 \times e^{-0.012 \times H}$ and $f_{Drop-tail}(H) = 87.68 \times e^{-0.020 \times H}$, respectively, where H is the ratio of HSTCP flows. Then, the ratio of occupancy $U(H)$ by HSTCP flows can be calculated from

$$U(H) = f(H) \times \left(1 - \frac{H}{100}\right). \tag{2}$$

By using Eq. (1) inversely, the target satisfaction rate $T(S_t)$ with achieving the target MOS value S_t is given by

$$R(S_t) = \frac{\log S_t - \log 1.416}{0.038}. \tag{3}$$

Therefore, the difference of the satisfaction ratio $\Delta(U)$ which should be controlled by the network support is

$$\Delta(U) = R(S_t) - U(H). \tag{4}$$

To achieve total $\Delta(U)B$ of throughput degradation of HSTCP traffic where B is the bandwidth of the egress link, we degrade the throughput with the ratio of $\Delta(U)$ equally for each HSTCP flow. From the bandwidth-delay product, the appropriate size of the congestion window can be obtained. We therefore calculate the target packet dropping probability p_d from

$$p_d = \frac{0.078}{\left(\left(B\frac{\Delta(U)}{N_H}\right)D\right)^{1.2}}, \tag{5}$$

where N_H is the number of HSTCP flows.

4.3 Effect of Network Support

We show the result of MOS values by applying the propose network support in Fig. 9. We set the target MOS value to be 3 (i.e., most people are satisfied). From this figure, the network support starts when the ratio of HSTCP flows exceeds 40%, and the proposed approach can achieve the MOS value more than 3 by increasing the ratio of HSTCP flows.

5 Concluding Remarks

In this paper, we have proposed a model for TCP version migration with the network support. In this model we have introduced MOS which is the subjective score with end users' point of view. Through experiments we have clarified the relation how the performance degradation make impacts to the subjective scores. Based on the results we propose a model of network support with keeping the target subjective score for TCP Reno users, and have shown that our model can achieve the advantage of HSTCP with minimizing complaints from TCP Reno users.

For future research topics, we need to improve the accuracy on the determination of packet dropping probability, and investigate the effect of the network support in drop-tail router cases.

References

1. Floyd, S., Ratnasamy, S., Shenker, S.: High speed TCP for large congestion windows. RFC 3649 (2003)
2. Wei, D., Jin, C., Low, S., Buhrmaster, G., Bunn, J., Choe, D., Cottrell, R., Doyle, J., Feng, W., Martin, O., Newman, H., Paganini, F., Ravot, S., Singh, S.: Fast TCP: From theory to experiments. IEEE Network (2005) 4–11
3. Kelly, T.: Scalable TCP: Improving performance in highspeed wide area networks. In: Proceedings of ACM SIGCOMM. (2003) 83–91
4. Chuvpilo, G., Lee, J.W.: A simulation based comparison between XCP and high-speed TCP. Laboratory for Computer Science Massachusetts Institute of Technology (2002)
5. de Souza, E., Agarwal, D.: A highspeed TCP study:characteristics and deployment issues. LBNL Technical Report LBNL-53215 (2003)
6. Rizzo, L.: Dummynet and forward error correction. Freenix 98 (1998)
7. Handa, Y., Minoda, Y., Tsukamoto, K., Komaki, S.: Measurements of utility for latency time in wireless service and its dependence on users' situation. IPSJ Journal (in Japanese) **2005**(47) (2005) 19–24

End-to-End QoS Monitoring Tool Development and Performance Analysis for NGN

ChinChol Kim[1], SangChul Shin[1], SangYong Ha[1],
SunYoung Han[2], and YoungJae Kim[2]

[1] National Computerization Agency Building 77, Mugyo-dong,
Chung-ku, Seoul, 100-775, Korea(Rep. Of)
{cckim, ssc, yong} @nca.or.kr
[2] Department of Computer Science and Engineering, Konkuk University, 1, Hwayangdong,
Kwangin-gu, Seoul, 143-701, Korea(Rep. Of)
{syhan, yjkim} @cclab.konkuk.ac.kr

Abstract. This paper intends to introduce the development of a terminal agent for QoS measurement that is suitable for an NGN environment, and to summarize the results of its performance test. Using the terminal agent for service quality measurement, it is possible to measure the QoE-based end-to-end service quality of voice phone, video phone, and VoD services. In addition, the terminal agent, installed in the user terminal (IP-based audio and video phones), as a software or hardware chip, measures the quality index for voice and video related multimedia services, such as R-value, MOS value, call success rate, one-way delay, jitter, packet loss, resolution. The terminal agent also applies the packet capturing method when using the actual service, and analyzes SIP, RTP, and RTCP protocol headers of the received packet. When the terminal agent was implemented in the IP video phone and compared with the performance of a general QoS measurement tool, there were barely any errors in the measurement value. It has also been identified that end-to-end service quality management, which is close to QoE, was possible.

Keywords: QoS, QoE, QoS Measurement, SQM(Service Quality Management).

1 Introduction

Due to rapid technological developments and the increasing diversity of user-based needs, current information & communication services are expected to evolve into a ubiquitous society, based on the Next Generation Network (NGN), in which communications will be enabled anywhere and any time. Accordingly, both the backbone and access networks have been highly advanced. In order to support application services with different traffic characteristics, next-generation protocols, such as QoS (Quality of Service), IPv6, security, and mobility protocols are being introduced. Individual services, including e-commerce, games, telephony, education, and broadcasting are being combined into, and developed as integrated services[11][12].

Led by the international organizations for standardization, such as ITU-T, IETF, and 3GPP, the standardization of QoS guarantee technologies, the basis of the NGN,

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 332 – 341, 2006.
© Springer-Verlag Berlin Heidelberg 2006

including the QoS Framework and DiffServ/MPLS/RACF[6][14], as well as multimedia service protocols, such as SIP[1][2], is underway. Since 2004, furthermore, domestic fixed & wireless carriers and equipment & terminal manufacturers have attempted to provide high-quality BcN trial services, such as the Broadband Convergence Network (BcN), voice & video phone, VoD, and IP-TV, through the BcN testbed projects. As existing individual wire & wireless broadcasting services, such as high-speed internet, PSTN, W-CDMA, and terrestrial broadcasting, have now been integrated into a single network, NGN needs to be able to support these individual services. Therefore, it is becoming increasingly vital to build a QoS network as well as support next-generation communication protocols, such as address (IPv6), security, and mobility protocols[15][16]. A QoS network provides different classes of service in terms of bandwidth, loss, delay, and jitter. It is therefore necessary to develop related technologies, such as QoS guarantee technologies (DiffServ, MPLS, and RACF), and a Service Level Agreement (SLA), as well as to support the standardization required. In particular, Service Quality Management (SQM) technology allows the core factors, such as QoS and SLA technologies, to be provided.

In existing networks, such as high-speed internet, service quality management has been carried out with a focus on the network through Probe, OAM, and common quality measurement tools. In an NGN environment, however, it is important to conduct end-to-end quality management by applying the degree of user satisfaction, based on Quality of Experience (QoE)[3], as well as for different network service classes. In order to provide end user SLA, the quality of actual service may be measured on a regular basis. Most existing service quality measurement tools, however, focus on network quality. Even the tool measuring end-to-end service quality is not able to provide a regular quality management service for all users.

Therefore, this paper intends to introduce the development of a terminal agent for QoS measurement that is suitable for an NGN environment, and to summarize the results of its performance test. Using the terminal agent for service quality measurement, it is possible to measure the QoE-based end-to-end service quality of voice phone, video phone, and VoD services. In addition, the terminal agent, installed in the user terminal (IP-based audio and video phones), as a software or hardware chip, measures the quality index for voice and video related multimedia services, such as R-value[4], MOS value[4], call success rate[2], one-way delay[7][8], jitter[7], packet loss[7][10] and resolution. The terminal agent also applies the packet capturing method when using the actual service, and analyzes SIP, RTP[5], and RTCP[5] protocol headers of the received packet. When the terminal agent was implemented in the IP video phone and compared with the performance of a general QoS measurement tool, there were barely any errors in the measurement value. It has also been identified that end-to-end service quality management, which is close to QoE, was possible. This is because the agent was easily implemented to SIP and RTP/RTCP-based service terminals in an NGN environment.

This paper comprises four chapters: Chapter 2 to the design of a terminal agent for NGN QoS measurement, Chapter 3 to the implementation and performance analysis, and Chapter 4 provides the conclusions and directions of future work.

2 Design of a Terminal Agent for NGN QoS Measurement

This chapter describes the design of a terminal agent for end-to-end QoS measurement for voice and video services in an NGN environment. Fig.1 below illustrates the processor architecture of the terminal agent for NGN QoS measurement.

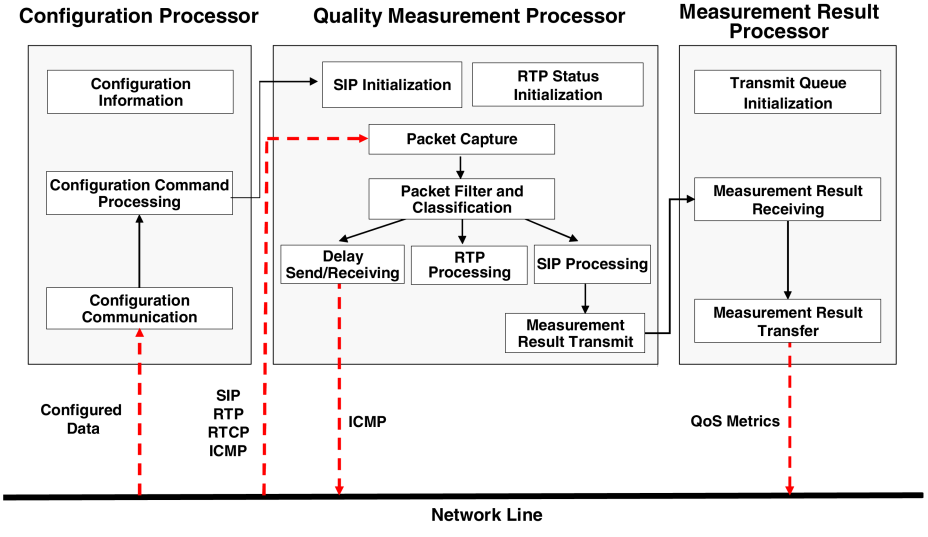


Fig. 1. Processor Architecture of the Terminal Agent for NGN QoS Measurement

The Configuration Processor makes it possible to search and change an environment variable for the activation of the terminal agent for NGN QoS measurement, while the Quality Measurement Processor captures SIP, RTP, and RTCP packets, analyzes the protocol header, and measures the quality indicators of NGN services (VoIP, video phone, and VoD). Finally, the Measurement Result Processor delivers these results to the QoS measurement server, using the RTCP XR protocol as defined in this paper. Each processor is defined as follows:

2.1 Configuration Processor

The Configuration Processor comprises a configuration initialization block, a configuration command processing block, and a configuration communication block. Each block is defined as follows:

- Configuration Initialization Block: This block sets the configuration information for the initial activation of the terminal agent for NGN QoS measurement. Table 1 illustrates the structure value for its initial configuration.

- Configuration Communication Block: This block receives Table 1-based configuration files from the QoS measurement server for the purposes of configuration of the quality measurement functions of the terminal agent, sending them to the configuration process block. The result from the configuration process block is then transmitted to the QoS measurement server.
- Configuration Command Processing Block: This block sets a set of command language from the configuration communication block to the terminal agent, then sending back the process result to the configuration communication block.

Table 1. The structure value of initial configuration

Data Name	Definition
SIP_POOL	SIP Session Table Size
RTP_POOL	RTP Status Table Size
DEVNAME	Network Device Name for Packet Capture
FILTER	Packet Filtering Tool
SIP_PORT	SIP UDP Port Number
RTSP_PORT	RTSP UDP Port Number
CONFIG_PORT	TCP Port Number for Configuration
GATHER_IP	QoS Measurement Server IP
GATHER_PORT	QoS Measurement Server Port Number
SERVICE_TYPE	Type of Terminal Service
QoS_INDEX	QoS Measurement Index List(R/MOS, Resolution, Delay, Jitter, Loss)

2.2 Quality Measurement Processor

The Quality Measurement Processor captures the packets that are required for quality measurement (SIP and RTP / TRCP), analyzes the protocol header, and measures the quality indicators (Connect Success Rate, One-Way Delay, Jitter, Packet Loss, R value, MOS, and Resolution). It comprises the following blocks, with each block defined as follows:

- SIP Session Initialization Block: This block creates a state management session pool for the SIP session to get the quality indicators, such as the connect success rate and resolution, organizes this into a linked list, and initializes the following main session information required for the measurement of quality indicators:

  { SIP Key (Source IP/Port, Destination IP/Port), Source DN (Dial Number), Destination DN, Call Phase (Fail, Success, In Process), Call Start Time, End Time, SDP Media Information }

- RTP Status Initialization Block: This block creates a state table for the RTP/TRCP packet process, organizes it into a linked list, and initializes the following main RTP status table information required for the measurement of quality indicators:

  { RTP Key (Source IP, Source Port, Destination IP, Destination Port), SSRC Field Value, RTP Start/Final Sequence Number, RTP Start/Final Time Stamp,

One-Way Delay (MIN, AVG, MAX) Value, Jitter Value, Total Received RTP Packet Count }

- Packet Capture Block: This block captures the packet from the terminal, using the packet capture library, and then sends it to the packet filter and classification block.
- Packet Filter and Classification Block: This block filters and classifies the packets required for quality measurement, and records the time stamp. After checking the port number, it sends the SIP packet to the SIP processing block, and the RTP packet to the RTP processing block.
- SIP Processing Block: This block receives the SIP packet from the packet filter and classification block, analyzes the SIP header and SDP protocol, measures the SIP session information (Source DN, Destination DN, Source IP, Destination IP, Call Start Time, and Call End Time), call success rate, and resolution, and records the result into the SIP session metrics of RTCP XR when the SIP session is terminated. The call success rate (%) is measured by a formula, {(number of total connected calls – number of failed-to-connect calls) / number of total connected calls * 100}, determining as SUCCESS where the status code of the SIP Response Message is 2xx or 3xx, and as FAIL where it is 4xx, 5xx, or 6xx after receiving the SIP INVITE message and analyzing the procedure up until 200OK and the ACK message. Resolution, a measurement of the total number of pixels displayed, is measured based on an SDP media attribute included in the SIP response message.
- RTP Processing Block: This block receives the RTP/RTCP packet from the packet filter and the classification block, analyzes the header, measures the quality indicators (voice and video services end-to-end one way delay, jitter, packet loss, R-value, and MOS value), saves them in the RTP state table, and outputs the measurements of the RTP quality metrics of the RTCP XR message at the SIP session is terminated. The quality indicator measurement method is defined as follows: One-way delay, an end-to-end one-way delay to the packets from the Send terminal to the Receive terminal, is measured by the formula {RTT/2} after getting the RTT with the reference of the DLSR and LSR field values of RTCP RR. Jitter, a variation of the end-to-end one-way delay to the packets from the Send terminal to the Receive terminal, is measured according to the formula, {J(i-1)+(Inter_arrival_jitter-J(i-1))/16}, where inter_arrival_jitter = l(R(i)-R(i-1) -(S(i)-S(i-1))l, based on the RTP packet arrival time. The packet loss rate is the rate of packet loss among the total packets of the actually transmitted data, after a normal call connection has been made. The quality indicator is measured by the formula, {(number of Send packets – number of Receive packets) / number of Send packets * 100}, with the reference of the RTP header sequence Number. R-value and the MOS value are measured using a end-to-end objective, subjective quality evaluation method based on E-model. For R-value is initially measured by applying the basic values of one-way delay, packet loss, and codec type, which measured through an RTP header analysis, to a formula suggested by the international standardization G.107. The MOS value is measured based on a converted formula. For variables other than the three items listed above, the basic values suggested by G.107 are used.

- Delay Sending/Receiving Block: When the application service in which the DLSR field value and the LSR field value in RTCP RR are not implemented, this is measured by the formula, {RTT/2}, after getting RTT by sending and receiving an ICMP message between the terminal agents for end-to-end NGN quality measurement.
- Measurement Result Transmit Block: This block receives the measurement result from the SIP and RTP process blocks, and stores it in the transmit queue.

Fig. 2 below illustrates a quality indicator measurement process after analyzing the packets from each block of the quality measurement processor.

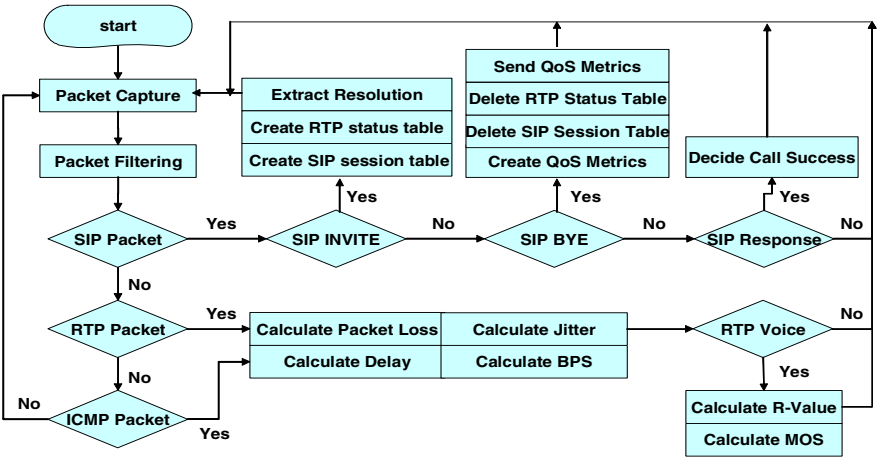


Fig. 2. Quality Measurement Processor Flowchart

2.3 Measurement Result Processor

The Measurement Result Processor receives the result from the Quality Measurement Processor, sending it to the quality measurement server. It comprises the Transmit Queue Initialization Block, the Measurement Result Receiving Block, and the Measurement Result Transfer Block. Each block is defined as follows:

- Transmit Queue Initialization Block: This block creates a Message Send Buffer to send the quality indicators from the QoS measurement processors to the QoS measurement server.
- Measurement Result Receiving Block: This block creates an RTCP XR message by reading the quality indicators stored in Transmit Queue by the Measurement Result Transmit block of Quality Measurement Processor, and send it to the Measurement Result Transfer Block. Fig.3. illustrates an RTCP XR message format, which is extended in this paper, to send the quality indicators to the QoS measurement server.

- Measurement Result Transfer Block: This block sends RTCP XR quality metrics from the Measurement Result Receiving Block to the quality measurement server using the TCP/IP protocol.

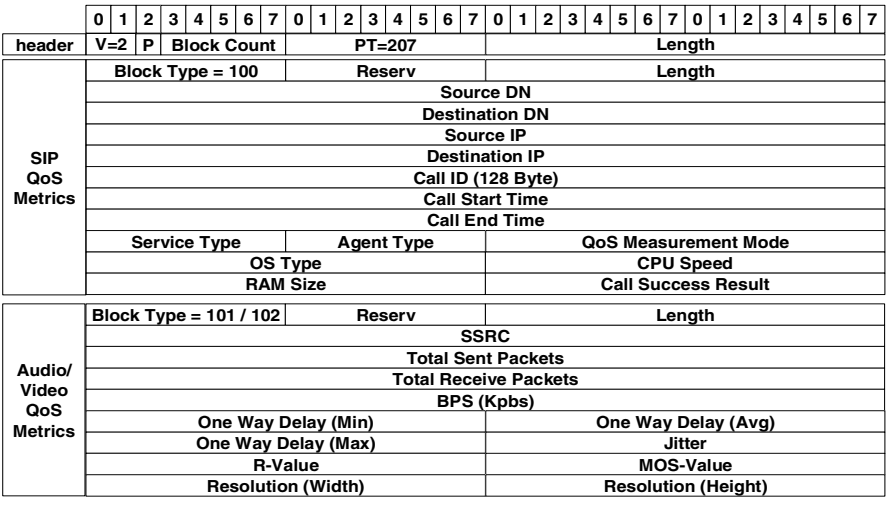


Fig. 3. QoS Metric extended the RTCP XR Message

3 Implementation and Performance Analysis

3.1 Result of Implementation

This paper has suggested the quality measurement terminal agent for an NGN environment, which uses the public packet capture library PCAP. For quality measurement functions were implemented using Windows CE or Linux-based Ansi-C language. In order to minimize the CPU load, specific modules were implemented based on the Posix Thread. After being ported to a Linux or Windows-based voice/video phone terminal, they are run as daemon type system services. Through SIP/SDP protocol analysis, a variety of information (i.e.: session information, type of service and terminal performance, call success rate, and resolution), can be measured. The quality indicators, (i.e.: R-value, MOS value, delay, jitter, and loss), can also be measured through an RTP/RTCP protocol analysis.

3.2 Test and Performance Analysis

This paper has tested the quality measurement function to evaluate the performance of the terminal agent for NGN QoS measurement. For the performance test, the terminal agent was ported to a real-time Linux-based video phone and compared with the common measuring instrument (QoS metrics) in terms of quality measurement results. Fig. 4 below illustrates the testbed for performance evaluation of the terminal agent for NGN QoS measurement:

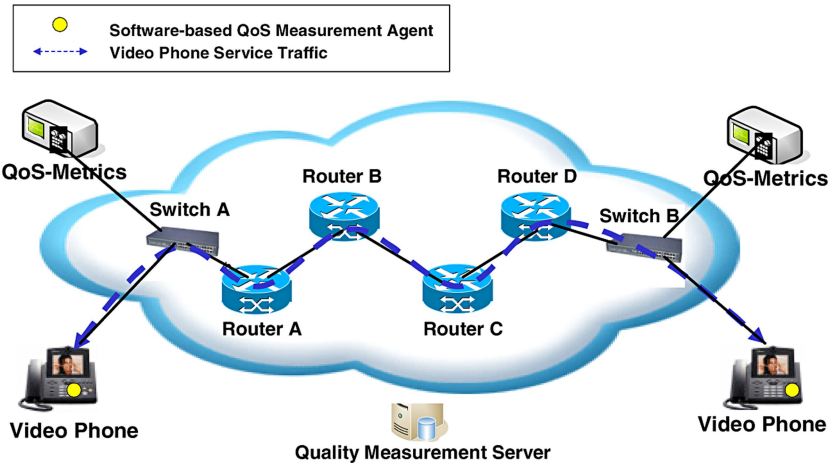
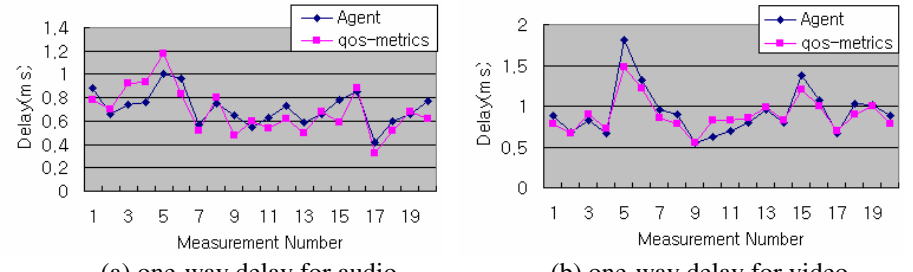


Fig. 4. Testbed for Performance Evaluation of the QoS Measurement Agent

The performance evaluation procedure of the NGN QoS measurement agent is defined as follows:

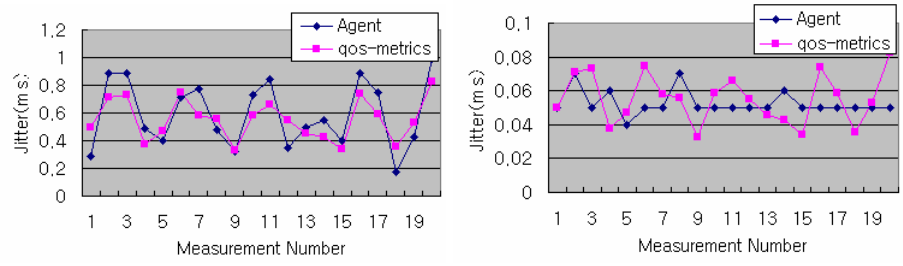
- With two video phones, make a five-minute telephone call twenty times. The NGN QoS measurement terminal agent analyzes the SIP and RTP/RTCP packets, measures the quality indicators, (i.e.: call success rate, packet loss, delay, jitter, R value, MOS value, and resolution), and sends the result to the QoS measurement server.
- Whenever a telephone call is made the packets on the video phone services are captured using a QoS Metrics (quality measuring instrument) simultaneously. The quality indicators, (i.e. packet loss, delay, jitter, R-value, and MOS value), are measured, and the error rate on the result is calculated.

Fig.5 and Fig.6 below illustrates the quality measurement result of NGN QoS measurement terminal agent on the video phone services. The result, (i.e. one-way delay and jitter), are measured by the terminal agent and QoS Metrics after being classified into voice and video. Fig.7 illustrates the Packet loss and R values, which are quality measurement result values.



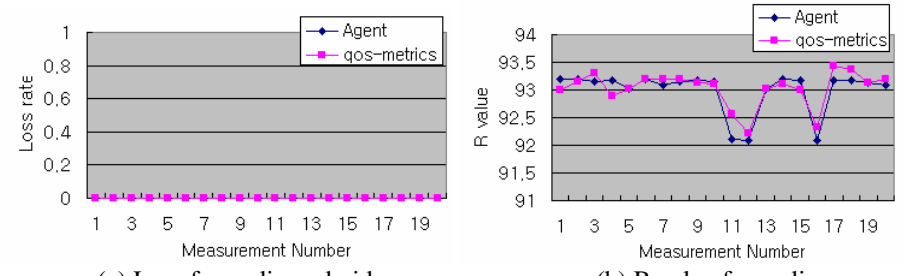
(a) one-way delay for audio (b) one-way delay for video

Fig. 5. Comparison of Measurement Result on One-way Delay



(a) jitter for audio (b) jitter for video

Fig. 6. Comparison of Measurement Result on Jitter



(a) Loss for audio and video (b) R value for audio

Fig. 7. Comparison of Measurement Result on Packet Loss and R value

According to the performance test result above, it appears that the one-way delay was less than 0.04ms on average, and that jitter was less than 0.05ms in terms of error span. No error was observed, however, in terms of packet loss. R-value was found to be less than 0.04 in terms of the error range. When the QoS measurement terminal agent was implemented in a video phone terminal, roughly 400Kbyte was consumed in terms of runtime memory use, while there was no influence determined on the basic video telephone service. This result vindicates the notion that the NGN QoS measurement terminal agent is reliable in terms of quality measurement performance, and suitable for QoE-based end-to-end service quality management in an NGN environment.

4 Conclusions

This paper has introduced the development of a QoS measurement terminal agent through which QoE-based end-to-end service quality management is made possible, and a performance test result is given. It has also demonstrated that the primary quality indicators, such as R-value, MOS value, call success rate, delay, jitter, packet loss, and resolution, may be measured by analyzing the packets on a regular basis whenever the service is used with a video phone in which the QoS measurement terminal agent is implemented. Based on the performance comparison with reliable common tools, it has additionally been confirmed that the error rate is approximately

the same. It has also been found that the agent is usable as a quality measurement tool for QoE-based end-to-end quality management in an NGN environment.

Notwithstanding these findings, further research needs to be conducted to compensate the function of the QoS measurement terminal agent through the development of video MOS, and to make the function accessible in a small terminal, such as a mobile receiver.

Acknowledgements. One of the authors of this work is supported by the second stage BK21.

References

1. Handley, H.Schulzrine, E.Schooler, and J.Rosenberg, “SIP: session initiation protocol”, RFC 2543, March 1999.
2. Wiley, "Internet Communications Using SIP Delivering VoIP and Multimedia Service with Session Initiation Protocol”, 2001.
3. "TMF 701: Performance Reporting Concepts & Definitions ver.2.0", TM Forum, Nov. 2001.
4. "ITU-T G.107: The E-model, A computation Model for use in Transmission Planning", 2003.
5. Addison Wesley, "RTP: Audio and Video for the Internet", 2003.
6. ITU-T Rec. Y.1711, Operation & Maintenance mechanism for MPLS networks, Nov. 2002.
7. V. Paxson, G. Almes, J. Mahdavi, “Framework for IP Performance Metrics”, RFC 2330 , May, 1998.
8. R. Koodli, R. Ravikanth, “One-way Loss Pattern Sample Metrics”, RFC 3357, August, 2002.
9. C. Demichelis, P. Chimento, “IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)” RFC 3393, November, 2002.
10. V. Raisanen, G. Grotefeld, A. Morton, “Network Performance Measurement with periodic streams”, RFC 3432, November, 2002.
11. EU Premium IP Project, http://www.cordis.lu/ist/ home.html
12. AQUILA Project, http://www.ist-aquila.org/
13. MESCAL, http://www.mescal.org/
14. TEQUILA, http://www.ist-tequila.org/
15. CAIDA, http://www.caida.org/
16. ChinChol Kim, SangChul Shin, Sunyong Han, “Architecture of end-to-end QoS for VoIP Call Processing in the MPLS Network, 2004

"P4L": A Four Layers P2P Model for Optimizing Resources Discovery and Localization

Mourad Amad[1] and Ahmed Meddahi[2]

[1] University of Bejaia, Algeria
mourad_amad@yahoo.fr
[2] GET/ENIC Telecom Lille1, France
Tel.: +33 3 20335562
meddahi@enic.fr

Abstract. Peer-to-Peer systems are based on the concept of resources localization and mutualization in dynamic context. In specific environment such as mobile networks, characterized by high variability and dynamicity of network conditions and performances, where nodes can join and leave the network dynamically, resources reliability and availability constitute a critical issue. To deal with this critical issue, we introduce a new concept and model called "P4L" (*four layers Peer-to-Peer model*) which define a novel P2P architecture, aims to improve: fault-tolerance, self- organization and scalability, with limited complexity while providing a rapid convergence for the lookup algorithm. The cost of "P4L" lookup is $O(\sum \ln(n_i))$ where n_i is the number of nodes on ring level i (*with maximum of 256 nodes in each ring*). "P4L" is efficiently adapted to the context where nodes join and leave dynamically and frequently. Each node maintains routing information of $2*O(\ln(n_i))$, where n_i is the number of nodes on one ring. Thus "P4L" is well adapted for terminals with limited resources such as mobile terminals. "P4L" is based on ring topology with each ring connecting "neighbouring" nodes in terms of physical and logical position. When "P4L" is combined with broadcast mechanism, the lookup process is significantly improved. The proposed model is evaluated and compared with Chord protocol, an extension is proposed to support IPv6.

Keywords: P2P, Routing,Complexity, P4L.

1 Introduction

Peer-to-Peer systems are distributed system without (*or with a minimal*) centralized control or hierarchical organization, where each node is equivalent in term of functionality. P2P refers to a class of systems and applications that employ distributed resources to perform a critical function such as resources localization in a decentralized manner. The main challenge in P2P computing is to design and implement a robust distributed system composed of distributed and heterogeneous peer nodes, located in unrelated administrative domains. In

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 342–351, 2006.
© Springer-Verlag Berlin Heidelberg 2006

a typical P2P system, the participants can be "domestic" or "enterprise" terminals connected to the Internet.

There are several definitions of the P2P systems that are being used by the P2P community [5]. As defined in [11], P2P "allows file sharing or computer resources and services by direct exchange between systems", or "allows the use of devices on the Internet periphery in a non client capacity". Also "it could be defined through three key requirements: **a)** they have an operational computer of server quality, **b)** they have an DNS independent addressing system" and **c)** they are able to cope with variable connectivity. Also, as defined in[2]: P2P is a class of applications that takes advantage of resources-storage, cycle, content, human presence-availability at the edges of Internet. Because accessing to these decentralized resources means operating in environment with unstable connectivity and unpredictable IP addresses. P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers [5].

In this paper, we introduced a new Peer-to-Peer model for resources discovery and localization, which can be classified as a structured P2P system. The nodes in this model are organized in four levels, each level is composed of several rings (*only the first level which is composed of one ring*). The main advantages of our proposed model are the rapid convergence of the lookup process, the number of active nodes can reach those of network Internet (scalability), and it provides an efficient mechanism for fault tolerance.

The paper is organized as follows : Section 2 gives a brief overview of P2P networks. Section 3 describes the lookup problem in P2P networks. The main characteristics of structured P2P systems such as distributed hash table are described in section 4. Section 5 presents and describes the proposed "P4L" model with a performance evaluation. Finally, we conclude and give some perspectives, particularly related to security aspects of P2P networks.

2 Related Work

Peer-to-Peer is relatively new in the areas of networking and distributed systems and services (*eg: Voice over IP*)[9]. P2P computing started to be a hot topic by the middle of years 2000. The different "generations" of P2P systems are characterized by transitions between generations motivated by different goals. In this section, we describe and comment the different generations of P2P network.

- **T**he first P2P generation like Napster file sharing application. The main contribution of Napster was the introduction of a network architecture were machines are not categorized as client and server but rather as machines that offer and consumes resources. All participants have more or less the same functionality. However, in order to locate files in a shared space, Napster provides a central directory. Napster is composed of two services : a decentralized storage service but with centralized directory service which can be a single point of failure.
- **T**he central coordination in the first solution leads to the transition to a new kind of P2P system, where the focus is on the elimination of the central

coordination. This generation started with Gnutella application [4]. The second generation systems solved the problem of the central coordination. However, the problem of scalability becomes more critical due to the network traffic load generated by the flooding algorithm for research or localization. Moreover, there is no guaranty to find a data item that exists in Gnutella system, due to the limited search scope.

– **I**n the third generation of P2P systems (*initiated by research projects such as Chord[10], CAN[6], Tapestry[12], Pastry[7]*), P2P systems are based on the Distributed Hash Table (*DHT*) to generate keys for both nodes and data. A node (*Peer*) in such system requires a unique identifier based on a cryptographic hash of some unique attribute such as its IP address. Nodes identifiers and key value pairs are both hashed to one identifier space. The nodes are then connected to each other in a certain predefined topology, eg: a circular space (*Chord*), d-dimensional cartesian space (*CAN*). Our proposed "P4L" model (*A four levels P2P model*) is derived from Chord and used a ring topology with hierarchical organization.

The common objective in all generation of P2P system is to optimize resources discovery and localization in a dynamic and heterogeneous P2P system. The next section describes this issue.

3 The Lookup Problem

The following example illustrates the lookup problem. Let's consider the case where a "publishers" inserts an item X, let's say a file or a resource in a dynamic system, while some consumers want to retrieve the item X at an other point or location.

In general the question is: when the publisher is connected to the system, how does the consumer find the location?

The third generation of P2P system is based on distributed hash table to generate a unique identifier for both node and resource, our proposed P4L uses DHT for only resource identifier. Because DHT is a key element of P2P networks (3^{rd} generation) and particulary in our contribution ("P4L"), we give a brief overview of DHT principles.

4 Distributed Hash Table (DHT)

A hash-table interface is an attractive foundation for a distributed lookup algorithms, as it places a few constraints on the keys structure or their associated resource. It also maps efficiently "keys" onto "value". The main requirement is that resource (*or data*) and nodes willing to store keys for each other (*key responsibility*) can be identified using unique numeric keys. This organization is different from Napster and Gnutella, which search for key words. The DHT implements one main operation: `Lookup(Key)` which yields to the node identity (*eg: IP address and port number*) currently responsible for the given key. A simple distributed storage application could use the interface as follows: The particular and unique name which is used to publish a file or resource is converted

to a numeric key using an ordinary hash function such as SHA-1 or SHA-2. Then a lookup (Key) function is called. Then the publisher sends the file or resource to be stored at the resulting node (*Data replication*). The requester who wants to access this file or resource, obtains the corresponding identifier, converts it to a key, call the `Lookup(Key)` function and ask the resulting node for retrieving a copy of the file or resource.

"P4L" is a scalable P2P protocol aims to optimize resources discovery and localization in a decentralized manner. It is based on DHT for resources identification and IP addresses for node identification. This model belongs to the third generation of P2P systems which are based on specific topology. "P4L" is organized on four levels, each one is composed of several rings. One of the most advantages of this proposed model is the rapidly convergence of the lookup process. Section 5 describes, analysis and compares "P4L" with others main P2P protocols.

5 P4L: Concepts and Principles

5.1 Motivations

The process of routing in P2P networks (*lookup data*) operates at application layer. The overlayed P2P network may lead to routing inefficiency as opposed to the routing service provided at the transport layer (*IP*). One of the P2P routing objectives is the minimization of the number of hops, for the localization of nodes storing data or resources. Our proposed "P4L" model is introduced for minimizing the number of hops for the localization process in a pure decentralized P2P networks. This approach allows for each node to maintain minimal information about others. So while "P4L" is a scalable protocol, it is also well adapted for terminals characterized by low resources such as mobile devices (*eg. PDA, mobile phone*). Each node maintains state of only $2 * O(\ln(n_i))$ where $n_i \leq 256$.

5.2 Resources and Nodes Identification in P4L

Each node in P4L is identified by a unique identifier n which is the i^{th} part of its IP address *(1 ≤ i ≤ 4)*, with i: the level which the node belongs to. The resources are also identified by a unique identifier generated by the distributed hash tables. Each resource key is composed of four parts (*a.b.c.d*). The example illustrated in *figure 1* shows that identifier of node x, associated to the IP address: 72.15.124.240 on ring level 1 is N72, while the node identifier associated to the IP address: 55.66.11.23 on ring level 2 is N66, this last does not belong to ring level one, as a node with identifier N55 already exists on the first level. The node with IP address: 50.31.10.111 on level 4 gets N111 like identifier as there are the nodes N50 on level 1, N30 on level 2 and the node N10 on level 3.

5.3 Resources Organization (P4L architecture)

Each resource with an *"a.b.c.d"* type identifier will be placed or located at the node with IP address *w.x.y.z* where w (respectively x,y,z) is the lowest value

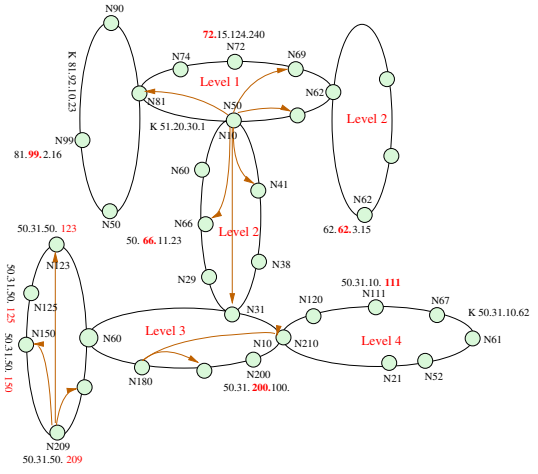


Fig. 1. "P4L" Architecture

greater or equal to *a* (respectively b,c,d). In *figure 1*, data associated to key K50.31.10.62 is placed on node N76 (with IP address 50.31.10.111) on level 4, as 50 (respectively 31, 10 and 62) is the lowest value greater or equal to 50 (respectively 31, 10 and 111).

The "P4L" model is based on structured and hierarchical rings. Each ring has 256 nodes (*maximum*). The "first" level ring contains nodes for which IP address are different in the first part, with no restriction in the others parts. From example (see figure 1), if a node n gets the IP address : 176.x.y.z, all other nodes with IP address type 176.a.b.c, do not belong to the node n ring. Each level has a maximum of 256^{i-1} rings (*i corresponds to the level number*). For nodes that have IP address like: *50.31.60.123, 50.31.60.125, 50.31.60.150 and 50.31.60.209* can belong to the same level (*level 4*). In this way, nodes are grouped together, based on their physical and logical proximity. At each level, nodes are organized based on their identifier. Each node maintains a routing table containing the IP address and port numbers of the "neighbouring" nodes. In each ring, nodes are ordered increasingly based on their identifiers.

5.4 Finger Table in "P4L"

Let m be the number of bits in the space of node identifiers on one level (m = *8 for 256 nodes*). Each node n maintains a routing table of at most m entries called the finger table. The i^{th} entry in the finger table of node n contains the identifier of the first node p that succeeds n by at least 2^{i-1} on the identifiers circle, where $1 \leq i \leq m$. We call node p the i^{th} finger of node n. A finger table entry includes both the "P4L" identifier and the IP address and port number of the relevant node. Figure 2 shows the finger table of one node belonging to two levels, and then has two identifiers: $N50$ on level 1 and $N52$ on level 2.

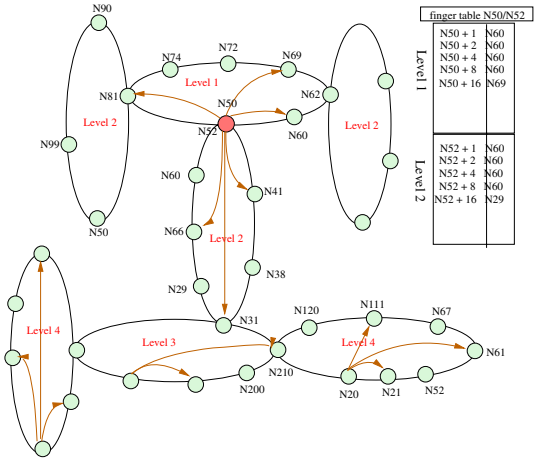


Fig. 2. Finger table in "P4L"

5.5 Lookup Process in "P4L"

Algorithm 1 gives a simple illustration for the lookup process in "P4L". For each level i ring, we use the i^{th} part of the data key. If the request succeeds on the ring k where the node requester belong, the cost of lookup algorithm is $O(\ln(n_i))$, with n_i the number of nodes in this ring ($n_i \leq 256$). In the case where the request fails on the first ring, the search or localization is done on ring level $k+1$ or $k-1$ in a deterministic manner, then the cost of the lookup process algorithm is $O(\sum(\ln(n_i)))$, where n_i is the number of nodes on ring level i, where the request gets success.

Algorithm 1. Lookup process (*P4L*)

Lookup (Key $c_1c_2c_3c_4$**)**
1: Begin
2: Locate the node X (*in the same ring*) of IP Address ($P_1P_2P_3P_4$) where P_i is the smallest value bigger or equal than c_i (using finger table).
3: **If** $\exists\ c_j$ where $c_j > P_j$ and $j < i$ **Then**
4: go to level $(i-1)$ and call lookup (Key $c_1c_2c_3c_4$) (if $i > 1$)
5: **Else**
6: **If** the data is present **Then**
7: loading data
8: **Else**
9: go to level $(i+1)$ and call Lookup (Key $c_1c_2c_3c_4$) (if $i < 4$)
10: End

5.6 Join Process in "P4L"

In a dynamic environment, nodes can join or leave at any time. The main challenge is to preserve the ability to locate and update every key in the network. For this, the "Bootstrapping" constitutes a vital core functionality, required by every Peer-to-Peer overlay network. Nodes intending to participate in such overlay network initially have to find at least one node that is already part of this network. Four solutions applicable for the Bootstrapping problem exist [1] and are resume as : Static Overlay Nodes-Bootstrapping servers, Out-of-bande Address caches, Random Address Probing or Employing Network Layer Mechanism. To reduce system complexity, we advocate Static Overlay Nodes-Bootstrapping servers for our proposed model P4L.

Finger table initialization: The i^{th} entry in the finger table at node n contains the identity of the first node s that succeeds n by at least 2^{i-1} on the identifier circle. If node n belongs to two levels, its finger table contains two parts, one for each level.

Transferring keys: When a node n joins the "P4L" system, there is a transfer of "responsibility" for all keys for which node n is now the successor. The node n becomes the successor only for keys that where previously responsible of the first node immediately following n. So n needs only to contact this one node to transfer responsibility for all relevant keys.

5.7 Leaving Process in "P4L"

When a node that belongs to one level leaves the system, some nodes must update their finger table. In other case (*the node that leaves the system belongs to two levels*), the stabilization algorithm (algorithm 2) is activated. For describes this process. We use the following notations: n : the identifier of the failed (leaving) node n and n_i, n_{i+1} : the two levels where node n belongs.

Algorithm 2. Stabilization process (*"P4L"*)

leave (n, n_i, n_{i+1})
1: Begin
2: **If** (n_{i+1} is empty) **Then**
3: update the routing table
4: **Else**
5: **If** $\exists$ (n', n'_i, n'_{i+1}) where $n'_i = n_{i+1}$ **Then**
6: **If** (n'_{i+1} = empty) **Then**
7: $n'_i = n_i$; $n'_{i+1} = n_{i+1}$;
8: update the routing table
9: **Else**
10: $n'_i = n_i$; $n'_{i+1} = n_{i+1}$;
11: update the routing table
12: **leave** (n', n_i, n_{i+1})
13: End

5.8 Fault Tolerance in "P4L"

In case of node(s) departure (*leave or failure*), the lookup process contributes to the robustness of the global "P4L" system. When a node leaves the "P4L" system, the "stabilization" algorithm (*see algorithm 2*) is executed for maintaining the correctness of successor's identifiers in the finger table. After a failure detection at node n by its neighbouring node, this last node invokes the join operation process, for localizing one node on "parent" level. Then the stabilization algorithm (*Algorithm 2*) is executed.

5.9 Lookup Acceleration in "P4L"

For accelerating the lookup process algorithm, a broadcast mechanism is used on each ring (*see Algorithm 3*). The number of messages generated does not have a significant impact on global performance, as each ring is limited by a maximum of 256 nodes. We assume that on each ring the nodes are completely connected. Thus the number of messages generated is $\sum(n_i - 1)$, where n_i is the number of nodes on each ring "covered" by the requests.

Algorithm 3. Lookup acceleration in "P4L"

Lookup (Key $c_1c_2c_3c_4$**)**
1: Begin
2: locate the node (*in the same ring*) of IP address ($P_1P_2P_3P_4$) where P_i is the smallest value bigger or equal than c_i (*by a simple broadcast message*).
3: **IF** $\exists$ c_j where $c_j > P_j$ and $j < i$ **Then**
4: go to level $(i-1)$ and call **lookup** (**Key** $c_1c_2c_3c_4$) (if $i > 1$)
5: **Else**
6: **If** the data is present **Then**
7: loading the data
8: **Else**
9: go to level $(i+1)$ and call lookup (key $c_1c_2c_3c_4$) (if $i < 4$)
10: End

5.10 "P4L" and IPv6

In order to supporte IPv6, we consider three cases for "P4L" protocol extension:

1. The "P4L" architecture described above is used (*based on a four levels topology*), with the 32 bits of the IP address for each ring.
2. We use the 8 bits portion of the total IP address for each ring, then the maximum of nodes for each ring is 256, but with a maximum of 16 levels. So the structure of format becomes $c_ic_2...c_{16}$.
3. The general case corresponds to the usage of k bits of the IP address for each ring, then the maximum number of levels is $128/k$. The first case is when $k = 32$ and the second case is when $k = 8$.

5.11 "P4L Properties"

"P4L" is characterized by a number of properties :

1. **Scalability:** Logarithmic growth of cost lookup with the number of network nodes even in "large scale" networks.
2. **Decentralization:** "P4L" is completely distributed, thus improving robustness of the global architecture (*each node is completely equivalent in term of functionalities*)
3. **Load balancing:** Distributed hash function spreads keys uniformly and evenly among nodes.
4. **Fault tolerance:** The lookup process is still active during node failure.
5. **Cost :** The cost related to the lookup algorithm in "P4L" is $O(\sum \ln(n_i))$. It is significantly better than Chord protocol as : already $\sum \ln(n_i) < \ln(n)$, where n is the number of nodes in the Chord system, and n_i the number of nodes on one ring of the "P4L" architecture.
6. **Flexibility:** Compatibility is guaranteed between IPv6 and IPv4 architecture, lookup, stabilization of system "P4L".

Table 1 resumes the main characteristics of "P4L" performance indications through a comparison with Chord. The two protocols are considered scalable and fault tolerant, but with a "better" cost lookup for "P4L".

Table 1. Chord vs "P4L" performance

	Scalability	**Fault tolerance**	**Cost lookup**
Chord	√	√	$O(\ln(n))$
P4L	√	√	$O(\sum \ln(n_i))$

6 Conclusion and Perspectives

Peer-to-Peer networks can be used for: improving communication process, optimizing resources discovery/localization, facilitating distributed information exchange [8]. Peer-to-Peer applications need to discover and locate efficiently the node that provides the requested and targeted service. "P4L" provides this discovery/localization service with a complete decentralized architecture by determining with efficiency the node responsible for storing the requested key's value. The node's identifier is simply derived from its IP address, while the resources identifier is generated by a hashing function (*with resource name as key*). One of the main characteristics of "P4L" model is the routing optimization at "IP" level, as it minimizes the number of hops for lookup process in the P2P network. Theoretical analysis shows that considering "P4L" model, the routing of resource lookup requests is significantly optimized. In a N-node network, each node maintains routing information for only about $2 * O(\ln(n_i))$, where n_i is the number of nodes on one ring with a maximum of 256. On rings belonging to "level 4", nodes are closed to (physically and logically). The use of Broadcast mechanism on this

topology can significantly and efficiently accelerate the lookup process. There are Broadcast-based and DHT-based algorithms for P2P protocols[3], "P4L" can be considered as a combination of Broadcast-DHT-based P2P protocol. P2P networks tend to become a key element for Internet communications such as legacy applications (eg: file sharing) but also for VoIP [9]. However, security aspects constitute a serious concern for P2P. (eg. in "P4L" context, a malicious node can present an incorrect view of "P4L" lookup process). For taking into consideration security aspect in "P4L", we are interested to combine and extend security protocols in the context of "P4L" for large peer group communications.

References

1. Curt Cramer, Kendy Kutzner, and Thomas Fuhrmann. Bootstrapping Locality-Aware P2P Networks. 2003.
2. C.Shirky. What is P2P..and what Isn't. *O' Reilly Network*, 2001.
3. Gang Ding and Bharat Bhargava. Peer-to-Pee file sharing over mobile Ad hoc Networks. *Proceedings of the second IEEE annual conference pervasive computing and communications Workshops (PERCOMW'04)*, 2004.
4. Gnutella. http://www.gnutella.com.
5. Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer to Peer conputing Survey. *HP Laboratories Palo Alto, HPL-2002-57*, March 2002.
6. Sylvia Ratnasamy, Paul francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content addressable network. *ACM SIGCOMM*, 2001.
7. Antony Rowstron and Peter Druschel. Pastry : a scalable, decentralized object location and routing for large scale peer to peer systems. *Proceedings of the 18 th IFIP/ACM international conference on distributed systems plateforms (Middelware 2001), Heidelberg, Germany*, November 2001.
8. Detlef Schoder and kai Fischbach. The Peer-to-Peer Paradigm. *Proceeding of the 38 th Hawaii International Conference on System Sciences, IEEE Internet computing*, 2005.
9. Kundan Singh and Henning Schulzrinne. P2P Internet telephony using SIP. Technical report, Department of computer Science, Columbia University, 2004.
10. Ion Stoica, Robert Morris, David Karger, M.Frans Kaashoek, and Haris Balakrishnan. Chord: A Scalable Peer-to-Peer lookup Service for Internet Application. *Proceedings of the ACM SIGCOMM'01, san Diego, California*, 2001.
11. Peer to Peer Working Group. Bidirectional Peer-to-Peer communication with interposing Firewalls and NATs . *Peer-to-Peer Working group, White Paper*, 2001.
12. Ben.y. Zhao, John kubiatowich, and Anthony D. joseph. Tapestry: an infrastructure for fault-tolerant Wide-area location and routing . Report No. UCB/CDS-01-1141, Computer Science Division, University of California, Berkeley, April 2001.

A Zeroconf Approach to Secure and Easy-to-Use Remote Access to Networked Appliances

Kiyohito Yoshihara[1], Toru Maruta[2], and Hiroki Horiuchi[1]

[1] KDDI R&D Laboratories Inc., 2-1-15 Ohara Fujimino-shi Saitama 356-8502, Japan
[2] KDDI Corporation, 3-10-10 Iidabashi Chiyoda-ku Tokyo 102-8460, Japan

Abstract. In this paper, we propose a new approach to secure and easy-to-use remote access to networked appliances (NAs). Based on the proposed approach, we develop a system in which servers in an Internet service provider network and a residential gateway (RGW) in a home network play an important role, including secure communication. The system allows us to access NAs, using mobile phone and PC anytime and anywhere. Our emphasis is that the system is based on zeroconf protocols including UPnP, and only by connecting an NA to a RGW enables secure remote access. Another emphasis is deployability. A RGW that has been widely deployed in home networks and always operates for VoIP call serves the zeroconf protocols. Interoperability with both UPnP compliant and noncompliant NAs, together available in the marketplace, is also considered. We deployed the system and conducted a field trial in which approximately 1700 users took part in for four months. Empirical results in terms of the zeroconf protocols will be shown with future directions.

1 Introduction

As we see always-on broadband access such as Fiber To The Home (FTTH), wide spread of mobile phones and emerging smart networked appliances (NAs), a variety of technologies have formed the basis of home networking. We can now have a connection anywhere from home and vice versa, as well as at traditionally limited universities or research institutes. This will steadily evolve into our new environment known as ubiquitous networking, which will support us by accessing everything connected together anytime and anywhere.

In such an environment, NAs would be a key driver for one of new multi-play services: remote access service. Typical NAs are a network camera, network attached storage (NAS) and network DVD player, which are abbreviated NW camera, NAS and NW DVD player in brief hereafter. With these NAs reachable from the Internet, we can check if a key to the front door is closed from a place away, by access to our NW camera facing the door, using mobile phone. Another scenario is remote recording reservation. We can make a recording reservation of a TV program, by access to our NW DVD player, using a mobile phone.

Some architecture and their associated protocols for NA remote access services have been developed and standardized. Typical protocols are based on Session Initiation

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 352 – 361, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Protocol (SIP). Unfortunately, they still stay at specifications and lab-scale prototypes and neither could they fully address the following technical issues: (1) configuration task, (2) usability nor (3) deployability, which will be described more in Sect.3.

For a solution to the issues, this paper proposes a new approach to secure and easy-to-use remote access to NAs. Servers in an Internet Service Provider (ISP) domain and a residential gateway (RGW) in a home network play an important role. The secure access is ensured by Security Architecture for Internet Protocol (IPsec) tunnel between an ISP domain and a home network. The easy-to-use access, which is one of our main contributions, is based on two zeroconf protocols: (1) Universal Plug and Play (UPnP) and (2) a new registration protocol. The approach has two stages: discovery and registration. In the discovery stage, a RGW discovers a new NA connected to a home network and monitors its connectivity using Simple Service Discovery Protocol (SSDP) and Simple Object Access Protocol (SOAP). In the registration stage, a RGW registers and updates a UPnP device description of connected NAs with servers using the new registration protocol. In addition, the zeroconf protocols are so designed that they can ensure the interoperability with UPnP noncompliant IP-based NAs. With the servers working as a rendezvous point, the proposed approach allows even Internet novices to access typical NAs available in the marketplace, using mobile phone and PC anytime and anywhere, with minimum user intervention.

An emphasis of the paper also lies in development and deployment of a system based on the proposed approach to show its practicality. We implement the zeroconf protocols in the form of RGW firmware and install the servers to work with the RGW, taking the current typical home network into account. For the practicality, we evaluate the proposed approach in terms of the two zeroconf protocols empirically through a field trial, in which approximately 1700 users took part in for four months, conducted as part of the deployment.

This paper is organized as follows: In Sect.2, we present an overview of a typical home network with FTTH connections and show technical issues for the NA remote access. We review the recent related work in Sect.3. In Sect.4, we propose a new approach to secure and easy-to-use remote access to NAs. In Sect.5, we describe implementation of the system, and evaluate the proposed approach empirically through the field trial.

2 Typical Home Network with FTTH Connections and Technical Issues for NA Remote Access

Figure 1 shows a typical home network, which we focus on throughout the paper, with fiber-optic network operator and ISP domains. Consumer premises equipment (CPE) devices including a media converter, RGW, PCs, and phone are connected in a tree to the RGW as its root. NAs are connected to the RGW in the same way. The RGW works as an IP router and has a Dynamic Host Configuration Protocol (DHCP) server for the home network. It has also Voice over IP (VoIP) capability. A home network is connected to the ISP domain, in which World Wide Web (WEB), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), SIP and Video on Demand (VoD)

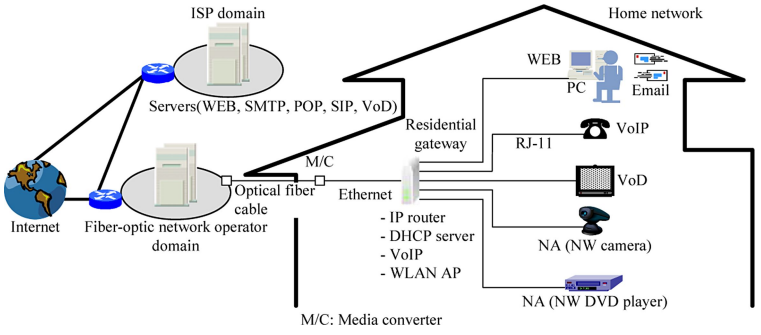


Fig. 1. Typical Home Network with FTTH Connections

servers are operated to serve such as triple play services via the fiber-optic network operator domain.

The following technical issues for the NA remote access should be addressed.

1. Reachability from the Internet to NAs should be ensured (Issue 1).
2. Communication with NAs should be authenticated and encrypted, to protect a network and privacy from unauthorized access by malicious or erroneous operation, for NAs are generally tied with individual activities (Issue 2).
3. An NA should be configured and discovered automatically, with minimum user intervention, to release home users from the complex configuration task and to inform users of the presence of dynamically connected NAs (Issue 3).
4. More usability by which an NA is given an associable name like ``Network Camera" instead of a usual form of ``http://192.168.0.1/xxx.cgi" should be achieved, so that users can select a connected NA with ease (Issue 4).
5. Architecture and protocols to realize the remote access should be subject to the current typical home network and to interoperability with NAs available in the marketplace, for the rapid deployment and acceptance (Issue 5).

3 Related Work

Research and development work on the NA remote access have been found. We summarize recent related work below and show that none of them alone addresses all issues in Sect.2 simultaneously.

The work propounded the use of SIP for the secure protocol of the NA remote access and discussed requirements and future challenges. Unfortunately, the work stayed at prospects and did not show how to address the requirements and challenges except Issue 1 and 2 in depth technically, while their contributions might be a few architectural frameworks and SIP message extensions.

Ubiquitous Open Platform Forum (UOPF) founded by Japanese companies has developed and standardized the UOPF protocol with its base architecture. The protocol

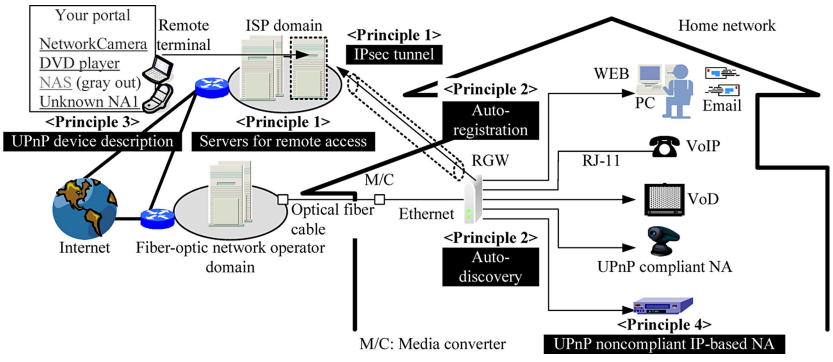


Fig. 2. Principles of Proposed Approach

addresses Issue 1, since it is also based on SIP and a user can access a target NA from the Internet using SIP as a signaling protocol. Network Address Translation (NAT) may be traversed with the aid of UPnP. The protocol meets Issue 2, for some extensions have been made to ensure the peer-to-peer secure communication using IPsec and Transport Layer Security.

Both protocols, however, could address none of the other issues due to their reliance on SIP. With respect to Issue 3, for each NA, a user should initially configure SIP Uniform Resource Identifiers (URIs) typically in the form of ``sip:xxx@isp.com'' to identify itself and gain access to SIP servers. As for Issue 4, due to the URIs typically given by an ISP as well as an email account, an NA cannot necessarily have an associable name, leading to poor usability. In terms of Issue 5, the protocol imposes SIP support on every NA. It is hard to find such an NA in the marketplace except for SIP phones as of writing, although the UOPF protocol provides specifications of a gateway for noncompliant NAs.

4 Proposed Approach

4.1 Design Principles and Assumptions

Design Principles. The proposed approach is designed based on the four principles as shown in Fig.2.

1. Cooperative architecture of servers and residential gateway
   Servers including a portal, Domain Name System (DNS), user database (DB) and Virtual Private Network (VPN) servers for the NA remote access are installed in an ISP domain. To address Issue 2, the access is authenticated by the servers and also encrypted by IPsec tunnel between the VPN server and a client on a CPE including a RGW. Main Issue 1 is solved as follows. A remote terminal including a mobile phone and PC first obtains web pages linking to NAs connected to a home network, from the servers. Next, the servers act as proxy for an access request, when the

terminal selects a target NA on the pages. The request is finally routed to the NA through the tunnel. Source of the pages is provided by the new protocols below.

2. New protocols for auto-discovery and auto-registration
   To tackle Issue 3, the RGW leverages DHCP, IPv6 Stateless Address Autoconfiguration (SAA) and SSDP M-SEARCH to auto-configure and auto-discover NAs that may be dynamically connected to a home network. The RGW obtains UPnP device descriptions from NAs, using SOAP after the discovery. In addition, a new auto-registration protocol based on Hyper Text Transfer Protocol (HTTP) by which the RGW registers the device descriptions with the servers is introduced to provide the web pages as described above. The detail of the protocol will be shown in Sect.4.2.

3. UPnP device description for more usability
   To address Issue 4, the web pages for the NA remote access is provided such that an associable string like ``NetworkCamera'' specified with the ``friendlyName'' tag in the UPnP device description is used as a link string. A URL specified with the ``presentationURL'' tag, which corresponds to a usual ``index.html'' of an NA, is used as a reference of the link. The string on a terminal may be grayed out and cleared, when a corresponding NA is disconnected.

4. RGW coping with UPnP noncompliant IP-based NA
   The RGW exploits Internet Control Message Protocol (ICMP) and ICMPv6 Echo requests to discover and register UPnP noncompliant IP-based NAs. The portal server allows a user to configure an associable name and TCP/UDP ports for those NAs.

With the four principles, we address Issue 5 in that 1) a RGW widely deployed in home networks and always operating for VoIP incoming call serves the discovery and registration, and 2) interoperability with both UPnP compliant and noncompliant NAs, available in the marketplace is achieved.

Assumptions. We put the following assumptions for the proposed approach.

1. The configuration of a RGW has been completed properly. Reachability from a home network to a subscribing ISP domain is ensured.
2. Servers for the NA remote access are installed. Moreover, a server that maintains versions of RGW firmware is set up in the same ISP domain. A RGW visits the server and checks if the current version is the latest, regularly or when the RGW (re)boots, so that it can keep its firmware up-to-date.
3. A user account and its associated password necessary to establish an IPsec tunnel are enrolled in the user DB by operators in advance.
4. A new version of RGW firmware for the proposed approach is released. It can establish an IPsec tunnel from a RGW to the VPN server automatically, and can then perform the auto-discovery and auto-registration. All required settings: IP addresses of the servers, a user account and password for the tunnel are preset, using a subscriber account as a key.

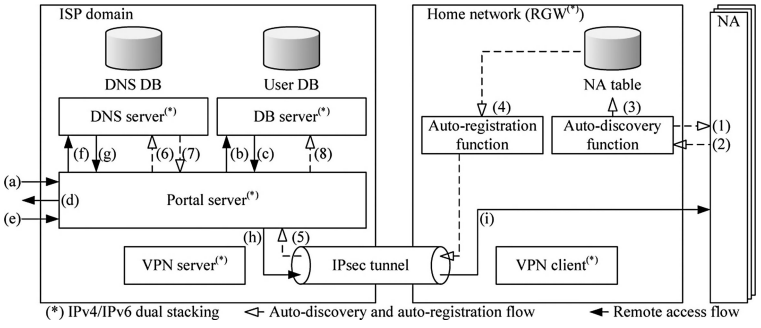


Fig. 3. Components and Flow of Proposed Approach

5. A RGW updates the current firmware with the above new version.
6. A user creates a user account and its associated password for the remote access, when the user first visits the portal server from a home network.

4.2 Components in ISP Domain and Home Network

Components in ISP Domain. Four server components: a portal, DNS, user DB and VPN servers constitute an ISP domain as shown in the left side of Fig.3.

The portal server operates as an HTTP server. It manages and provides web pages linking to NAs connected to a home network for each user, and processes the NA remote access requests. It generates a pair of a Fully Qualified Domain Name (FQDN) and its associated IPv4 or IPv6 address for each NA, on receiving a registration request from a RGW, to resolve duplication caused by the same IPv4 address space among users. The server allows a user to customize preferences: to create, delete and change a user account and password for the remote access, to control access to NAs based on the user account and to configure an associable name and TCP/UDP ports for UPnP noncompliant IP-based NAs. With the DNS DB, the DNS server is responsible for the registration of pairs of an FQDN and its associated IP address, when it receives a request from the portal server, as well as the original name-to-address resolution. The user DB server manages user related information: a subscriber account, pairs of a user account and password to establish an IPsec tunnel and for the remote access, and part of the source of the web pages. The VPN server establishes and releases IPsec tunnels based on requests from a VPN client in a home network.

Components in Home Network. Three components: auto-discovery function, auto-registration function and VPN client are shown in the right side of Fig.3.

With the auto-discovery function, a RGW discovers an NA, using SSDP M-SEARCH, ICMP and ICMPv6 Echo request. The three types of requests are sent in order on a regular basis. When the RGW receives an M-SEARCH response, it obtains a UPnP device description of a corresponding NA using SOAP. The RGW manages a Media Access Control (MAC) address, IPv4 address, IPv6 address (optional), deviceType, presentationURL, uuid and friendlyName, where the last four items are

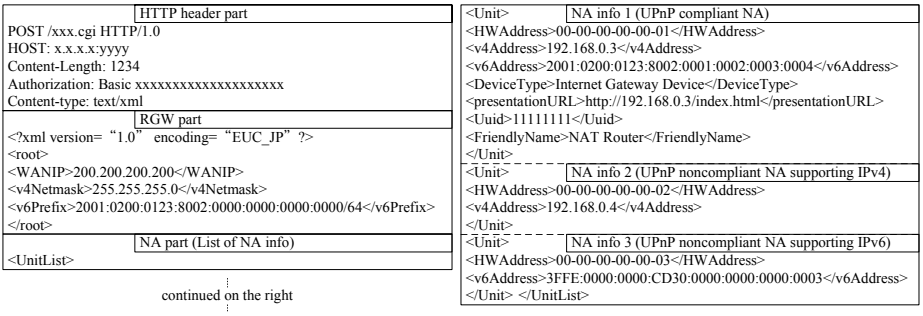


Fig. 4. Example of Auto-Registration Protocol Payload

specified in device descriptions, in the form of a logical NA table. The RGW monitors NA connectivity by regular M-SEARCH request, and updates the NA table. The RGW deletes an entry of the NA table if it does not receive a specified number of consecutive responses from a corresponding NA, leading to clearance from a screen via gray out. In terms of ICMP, addresses to which a RGW sends the requests can be limited with a start address and a number of addresses, such as ``192.168.0.2" and ``6", to avoid bandwidth consumption. With respect to ICMPv6, a RGW first sends a request to the link local multicast address ``FF02::1". When the RGW receives a reply, it sends a subsequent request to a global address that is generated based on SAA. On receiving an ICMP or ICMPv6 Echo reply, the RGW stores a MAC and IP address of the NA, and monitors in the same way as M-SEARCH request. Two or more entries may be stored for an NA, if the NA operates on IPv4/v6 dual stack and replies to both ICMP and ICMPv6 Echo requests. Such duplication will be resolved on the portal server, using MAC address as a key.

Auto-registration function registers the NA table with the portal server on a regular basis, using a newly introduced protocol based on HTTP. An example of the protocol payload is shown in Fig.4. The RGW part includes Wide Area Network (WAN) addresses of the RGW. The NA part is defined as a list. Each element corresponds to an entry of the NA table. The list can contain an arbitrary number and type of elements, being independent of UPnP compliance.

The VPN client requests the VPN server to establish and release an IPsec tunnels. It sends keep-alive messages for the tunnel maintenance.

4.3 Auto-Discovery, Auto-Registration and Remote Access Flow

We describe the two flows of the proposed approach with Fig.3, to show how it allows for the secure and easy-to-use remote access.

Auto-Discovery and Auto-Registration Flow. When a user connects an NA to a RGW, a RGW configures link-local IP settings. The RGW discovers an NA (Fig.3(1)(2)), and creates an entry for the connected NA in the NA table (Fig.3(3)). Asynchronously with the discovery, the RGW registers the NA table with the portal server through the IPsec tunnel (Fig.3(4)(5)). The portal server receiving the

registration request generates pairs of an FQDN and an IP address for NAs, and registers the pairs with the DNS server (Fig.3(6)). After successful registration (Fig.3(7)), the portal server stores the up-to-date information including the payload as shown in Fig.4 in the user DB server (Fig.3(8)).

Remote Access Flow. When the portal server receives a remote access request from a user (Fig.3(a)), it authenticates the user with the aid of the user DB server (Fig.3(b)). Success of authentication (Fig.3(c)) provides the user with web pages for the remote access via the portal server (Fig.3(d)). When the user selects a target NA (Fig.3(e)), the portal server creates a query for name-to-address resolution to the DNS server (Fig.3(f)). The request is routed to the resolved address (Fig.3(g)) through the IPsec tunnel (Fig.3(h)).

5 Implementation and Empirical Evaluations

5.1 Implementation

We implement a system based on the proposed approach and describe a brief overview of the system below, including some restrictions on the implementation.

1. We implement the servers on separate hardware, using general-purpose server. The servers are connected with 100Base-TX Ethernet in adjacent LANs, while we implement the DBs, using a single storage (4.5Tbyte in physical) that is connected to its associated server with fiber-channel. The specification of the portal server in terms of a quadruplet (CPU(GHz), memory(Gbyte), HDD(Gbyte), OS) is (UltraSPARC IIIi 1.28 × 2, 2, 73.4 × 2, Solaris9.9).
2. We implement a new version of RGW firmware for the proposed approach. The target RGW is models that have already operated in home networks.
3. We implement the VPN client software that runs on a user PC, due to some models that are restricted in processing capability. In this case, to establish an IPsec tunnel, a user should enter an account and password on a PC.
4. Interoperability with NAs is ensured, such as NW camera, NAS and NW DVD player, shipped from multi-vendors and available in the marketplace.

5.2 Empirical Evaluations

We deployed the system in Sect.5.1 with which a commercial FTTH network for a triple play service was overlaid, to conduct a field trial for four months. We provided the firmware to approximately 1700 users who were the FTTH subscribers and applied for the trial. We recommended the use of NAs that were ensured the interoperability to the users.

The amount of messages for the auto-discovery by the system in 5.1 will be shown first, in order to evaluate the traffic overhead. Part of server activities in logs during the trial will be shown next, so as to evaluate the overhead of the auto-registration and to estimate the scalability towards commercial deployment. Some discussions to improve the system will be made after each evaluation.

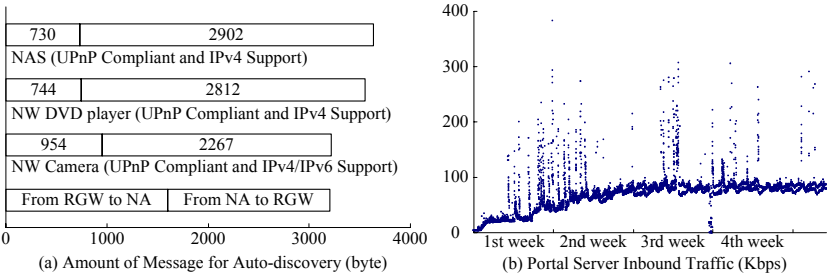


Fig. 5. Measurement Results of System during Field Trial

Evaluation of Auto-Discovery. Figure 5 (a) shows the amount of messages for the auto-discovery, when we connect each NW camera, NAS and NW DVD player to the RGW with 100Base-TX Ethernet. Due to a device description, the amount of messages to RGW is more than that from RGW for every NA. The amount of messages to the NW camera is more than those of the other NAs, for an ICMPv6 Echo request and reply. The total management traffic is 83.5bps (= (730+2902+744+2812+954+2267)*8/60) when the monitoring interval is 60 seconds, which was the same as in the trial. This is negligibly small, for it is much less than bandwidth of a typical home network: 100Mbps. We can shorten the interval to provide users with a more consistent view of dynamically connected NAs, since the response time from an M-SEARCH request is at most 421.6msec. In addition to such regular discovery, we will be able to utilize ``ssdp:alive" and ``ssdp:byebye" message, although listening to such message required more processing capability from the RGW than that at the design phase.

Evaluation of Auto-Registration. Figure 5 (b) shows the inbound traffic of the portal server monitored every five minutes for the first month. It includes only the traffic used for the remote access. At least around 80Kbps traffic arriving constantly from the third week can be seen for the auto-registration, where its time interval was 10 minutes. The amount of messages for the auto-registration per user is estimated at approximately 14.1Kbyte (=80*10*60*/(1700/4)/8), if we can assume one fourth of users had already joined. This is acceptable, when we recall an ISP-style email service. A huge number of users are sending pile of emails of the size that is typically larger than that, due to rich contents attached.

An unavailable time at the end of the third week was caused by a planned network construction. Redundant configuration in the ISP domain in terms of both hardware and software will be required toward the scalable commercial deployment, to avoid such a single point of failure.

6 Conclusions

This paper proposed a new approach to secure and easy-to-use remote access to networked appliances (NAs). We show five technical issues for the NA remote access. Lack of a protocol addressing all issues simultaneously motivated us to design a new

approach based on (1) cooperative architecture of servers and residential gateway (RGW), (2) new zeroconf protocols for auto-discovery and auto-registration, (3) UPnP device description for more usability and (4) RGW coping with UPnP noncompliant IP-based NA. Elaborated description of the components that constitute the system was shown. The proposed approach allows for secure access to NAs available in the marketplace, using mobile phone and PC anytime and anywhere, with minimum user intervention. We implemented a system based on the proposed approach, and evaluated the zeroconf protocols empirically through a field trial for four months. We released a new RGW firmware to approximately 1700 FTTH subscribers. The overhead of the zeroconf protocols were shown acceptable in terms of the amount of messages.

We believe the proposed approach is the best practice as of writing and will confront the dilemma of the chicken (seeds) or the egg (needs), as often appears in service-oriented domains. The commercial deployment and the integration with the advantage of SIP based protocols will be future directions.

Acknowledgment

We are indebted to Dr. Shigeyuki Akiba, President, Chief Executive Officer of KDDI R\&D Laboratories Inc., for his continuous encouragement to this research.

References

1. Moyer, S., Marples, D., Tsang, S.: A Protocol for Wide-Area Secure Networked Appliance Communication. IEEE Communications Magazine (2001) pp.52–59
2. Moyer, S., Marples, D., Tsang, S., Ghosh, A.: Service Portability of Networked Appliances. IEEE Communications Magazine (2002) pp.116–121
3. Ubiquitous Open Platform Forum (UOPF): UOPF Protocol. (2004) http://uopf.org/en/ (URL available on April 2006).
4. Rosenberg, J., et al.: SIP: Session Initiation Protocol. IETF RFC 3261. (2002) 5. UPnP Forum: Universal Plug and PlayTM Device Architecture. (2000)
5. World Wide Web Consortium: SOAP Version 1.2. (2003)
6. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration. IETF, RFC 2462. (1998)
7. Conta, A., Deering, S., Gupta, M.: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. IETF, RFC 4443. (2006)

A Test Method for Base Before Service (BS) of Customer Problems for the NeOSS System

InSeok Hwang[1], SeungHak Seok[1], and JaeHyoong Yoo[1]

[1] KT Network Technology Laboratory,
463-1 Junmin-dong, Yusung-gu, Taejeon, Korea
{ishwang1, suksh, styoo}@kt.co.kr

Abstract. As a network & service provider, KT has long felt the necessity of integrating many of its OSS (Operation Support System) systems, and in 2003, KT finally developed a new system to serve its various maintenance activities. This system is called the NeOSS (New Operations Support System) System, which uses XML (Extensible Markup Language), EAI (Enterprise Application Integration) and Web service based on Microsoft's .NET platform technology. This paper shows the system architecture of the NeOSS-ADM(Access Domain Management), which is going to be developed using the .NET environment, service maintenance framework and the method of maintenance for NEs (network elements) for base BS. And this system will give more high quality service to customers and reduce the customer's complaints.

Keywords: NGOSS, NeOSS, NeOSS-ADM, Proactive Service.

1 Introduction

Korea is the country with the highest distribution rate for high-speed internet service in the world, and in recent years, its customer base is becoming saturated. The number of high-speed internet customers in Korea has grown 12.26 million (05.12). KT alone accounts for more than 6 million customers. Thus, KT is confronted by keen competition due to the saturation of customers and the emergence of diverse service providers that offer high-speed and broadband network services. In addition, customer requests have become more diverse, from service orders to complaints related to fault resolutions, quality of service, cost, rapid processing, and others.

In recent years, telecommunication services are being provided through complex networks instead of single networks, particularly the various internet services. Thus, many companies have developed many kinds of new services in the desire to obtain larger market shares. To give the general public a high-speed internet access service, KT supports various access technologies, including PSTN (Public Switched Telephone Network), ADSL and VDSL, in order to provide broadband services through copper wire. KT had also established an evolution plan for optical fiber access networks. In accordance with this plan, KT developed and deployed the FLC (Fiber Loop Carrier) –A and -B systems for business applications (FTTO) and the FLC-Curb(C) and –Dense(D) systems for densely populated areas (i.e. apartment complexes) and small business applications (FTTC).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 362 – 371, 2006.
© Springer-Verlag Berlin Heidelberg 2006

KT also recently developed an AGW (Access Gateway) for the BcN (Broadband Convergence Network). In the near future, KT will also deploy a soft-switch for the BcN. The FLC-C, -D and AGW systems are able to connect to network switching systems that support the ETSI V5.2 standard.

KT also needed a new paradigm for its OSS system because of the continuous emergence of new products (Internet services) and extensive customer needs. In particular, customers' satisfaction level for service quality is much higher than before. Thus, KT concluded that these issues cannot be resolved without basically changing its operational management system. Therefore, KT developed the NeOSS system to establish a new, professional and flexible architecture system. The NeOSS system is an integration of the telecommunication management system and rebuilding system per facility part with the management process TMN (business management, service management, network management, equipment management) based on other coupled systems. The NeOSS manages all of the company's information relating to quality, facility, traffic, customer service, and others.

2 NeOSS Structure

Our system's goal is to support a one-stop fault management scheme and building system. We can check where the NeOSS-ADM is on the NGOSS eTOM model and how we can develop our system to co-work with any other system or facility.

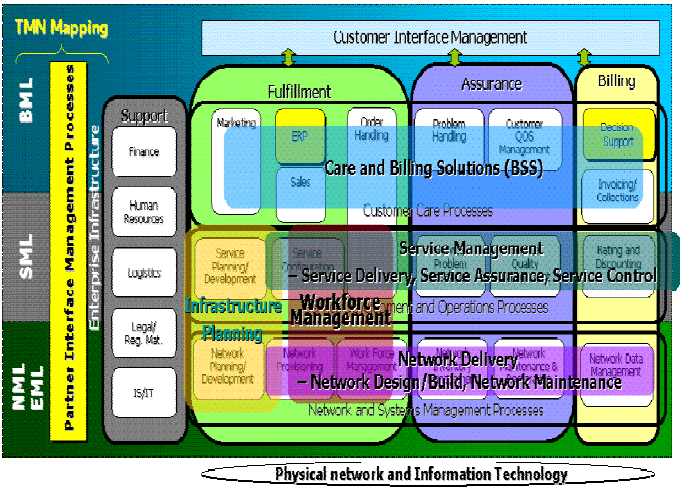


Fig. 1. The NeOSS Architecture is harmonized with the NGOSS Framework and eTOM

The NGOSS is driven by TMForum for the standard business solution framework of the next operation support system (OSS)/business support system (BSS). The main contents of the NGOSS are business process modeling, design process, choice technology, public data modeling and implementation. eTOM is the general business

activity map that service providers must provide and the business process model enterprise business process framework that service providers must request. The eTOM model decides the borderline of software systems and components supplied by the OSS/BSS. A concluding remark is that the eTOM model provides liaison with future NGOSS and improves the relationships of businesses under the more complex circumstance of e-business.

We applied a CBD (component-based development)-based architecture to minimize the dependency between components and also increased the reusability to reduce development costs. The NeOSS-ADM locates the SML layer and "Service Assurance" part in the eTOM business process model.

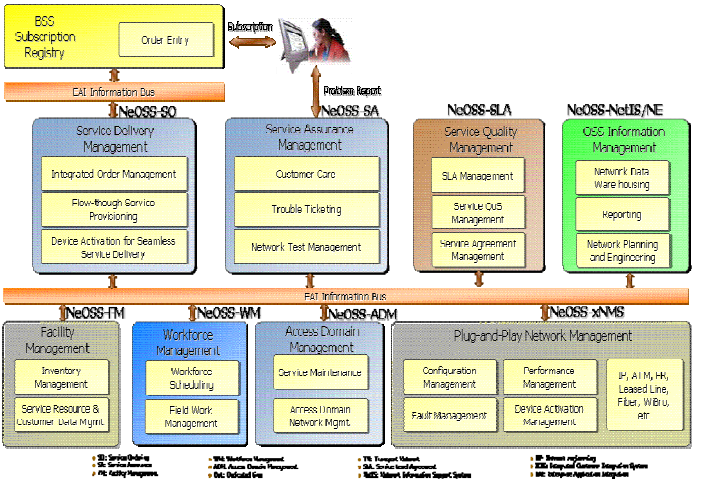


Fig. 2. The NeOSS system

The NeOSS consists of SA (Service Assurance), which manages the reception of customer concerns, SO (Service Order), which manages customers' installation orders, FM (Facility Management), which manages the network facility and customer database, ADM (Access Domain Manager), which manages the domain of the access network, WM (Work Management), which manages outside work, SLA (Service Level Agreement), which manages the VOC (Voice of Customers), NetIS/NE (Network Information Support System), which supplies statistical data and information, and xNMS (xDSL NMS), which manages the network facility. The NeOSS-ADM consists of service orders for customer installations and real-time connection with network elements for fault management & service order.

The following are the technologies applied to the NeOSS: the NeOSS consists of five modules that use the EAI Workflow concept based on the COM+ (DLL) Services Call Method. The COM+ (DLL) Services Call Method is based on the .Net Framework 1.1, which is a state-of-the-art technology. EAI uses BizTalk2004 as well. For the database systems, a great part of the NeOSS DB is constructed on the SQL2000 64-bit version. The NeOSS arranged the standard inter-operation methods

into a few and simple categories and chose them to achieve inter-operability with legacy systems.

Most Service Providers focus on one or two offerings when introducing new services. As a result, similar functions are reproduced by various OSS/BSS applications. This kind of limited scope results in offerings that are very poorly coordinated and have few synergies. They also take longer to build new services (almost starting from beginning each time) and new services are expensive to introduce. To solve this problem, we defined the principle and mechanism of inter-operation with other OSS/BSS. The inter-operation mechanisms are as follows:

- Web Service (Standard)
- Web Service (Protocol Adapter)
- EAI (Standard)
- EAI (Protocol Adapter)

Protocol Adapter mechanisms were applied for inter-operating with legacy systems such as various NMS or BSSs. In this mechanism, we provide adaptors such as the socket-based method, database-based method, XML-based method and others. Standard mechanisms were applied for inter-operating between the NeOSS subsystems, such as NeOSS-SO, NeOSS-SA, NeOSS-ADM, NeOSS-WM and NeOSS-FM.

3 NeOSS-ADM System Architecture

The NeOSS-ADM consists of NeOSS-ADM (.NET) and NeOSS-ADM (UNIX). The operator of the NeOSS-SO (Service Order) and NeOSS-SA (Service Assurance) receives the customer's new order installation and trouble report for a telephone or an Internet service. Thereafter, the control of the work order or trouble report is passed over to the NeOSS-ADM to test the condition of the subscriber's line or to test the Internet line performance using the TU (Testing Unit). Then, the NeOSS-ADM sends the test result to the NeOSS-SO or the NeOSS-SA.

The NeOSS-ADM (.NET) is located in the central office and plays the role of SML (Service Management Layer) and NML (Network Management Layer) as the TMN (Telecommunication Management Network). The NML gives a network-level view to facilitate network planning and fault correlation. It provides network configuration, fault and performance windows for networks. The SML of the NeOSS-ADM provides the service provision, service maintenance and network management. It includes customer complaint-handling and customer provision application.

The NeOSS-ADM (UNIX) has communication services that include SNMP/TCP/HTTP services. The NeOSS-ADM (UNIX) communicates with FLC systems through TCP, ADSL by SNMP and HTTP. The NeOSS-ADM (.NET) communicates with the NeOSS-SA and NeOSS-SO through SOAP/XML. The NeOSS-ADM (.NET) has a web-gateway that enables customers to report troubles in the system through the Internet. NeOSS-ADM (.NET) operators can connect directly to the system or through the Internet. NeOSS-ADM operators in the transmission room, exchanging rooms and testing room of the local office can connect to the system through the web-gateway.

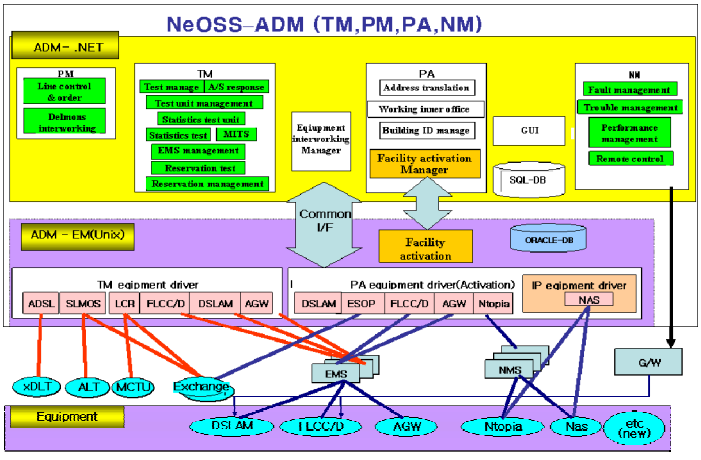


Fig. 3. The NeOSS-ADM system and inter-operation with other systems

In Figure 4, we show part of the web service (protocol adapter) for NeOSS's inter-operation with other systems. The web service method consists of adapters, such as socket, XML-RPC, HTTP and Oracle Database adapters. Through a sync or Async way, the web services also decide on whether or not to use the EAI bus (the EAI bus basically uses the Async way). Figure 4 shows its inter-operation with other systems.

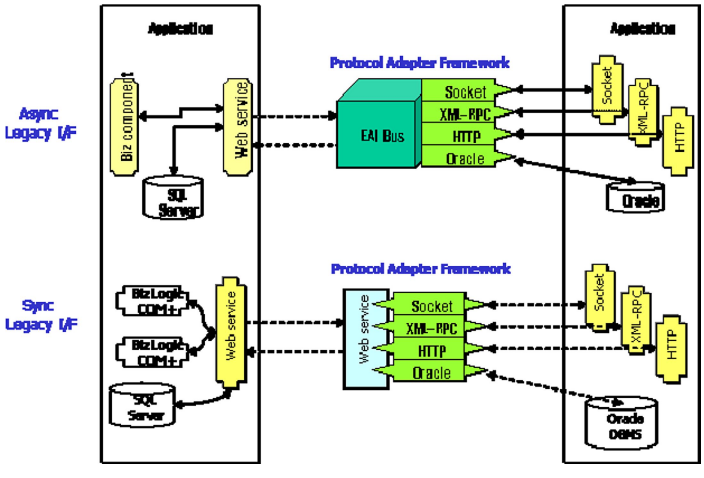


Fig. 4. Inter-operation with other systems

KT started to develop the NeOSS (New Operation Support System) to reinstall the communications net and network management processing module and provide a new and advanced flexible architecture system in 2003. The NeOSS system underwent

operation and evaluation in Chungcheong province (it has 44 offices) starting September 14, 2004, and the NeOSS system is now being operated all over the country.

In this paper, we focus on the NeOSS-ADM-TM (Trouble Management) system. It charges the fault management processing, which tests with TU (Testing Unit), or controls the network element EMS (Element Management System). The NeOSS-ADM-TM system provides integrated maintenance services for customers' trouble reports about POTS (Plain of Telephone Service) and high-speed Internet service.

Maintenance activities should be performed after service is provided and activated. ALTC (Automatic line tester controller) and MCTU (Metallic Cable Test Unit) equipment have been charged with those missions, which is, receiving customer complaints about telephone services and carrying out testing of the subscriber line and responding to them. Other EMS or NMS (Network Management System) have been charged with missions such as receiving customer complaints about high-speed internet services and carrying out testing of the Internet service line and responding to them. The NeOSS-ADM-TM offers management of customers' line status fault and reception of customer complaints over a copper access network and a broadband access network (an optical access network, an ADSL (Asymmetry Digital Subscriber line)).

When customers experience problems, they can contact a KT customer service center using the telephone or Internet, and operators can respond to customers' trouble reports and complaints using PCs connected to the NeOSS Web Server. In addition, the NeOSS provides proactive maintenance services, that is, before customers can report their complaints, they can be notified of status reports and planned maintenance schedules using the function of service-affected customer identification, performance/fault trend check, performance/fault analysis and QOS assurance. The next section will explain the proactive method of the NeOSS-ADM-TM in fault management processing.

4 METF(Management EMS Test Function)

The NeOSS-ADM-TM can provide network element management services, that is, real-time status report data from the access network, which consists of DSLAM, FLC-C/D, AGW and Exchange, and offer the statistical data to analyze the status of the EMS. The METF (Management EMS Test Function) consists of a variety of functions (real-time monitoring, test-rate management, etc.) for controlling the EMS of each network element. We can always understand the state of every EMS in real-time using this function.

4.1 EMS Monitoring

The EMS Monitoring function monitors the real-time state of the EMS for each network element. We send the specific message, which are hello and ping, to the EMS. The ping message checks the network status from the NeOSS-ADM-TM to the EMS. If we get a result of fail, we conclude that the network is not connected from the NeOSS-ADM-TM to the EMS. Otherwise, we conclude that there is no problem with the network status.

The hello message checks the specific processing module of the EMS, which connects to the NeOSS-ADM-TM system's processor. If we get a result of fail, we conclude that the EMS' process is dead. So, we call the manager who manages the EMS system. In addition, we can set the period test for Hello & Ping. For example, we can select a period of 5 or 30 seconds, and then the NeOSS-ADM-TM system sends the Helloand Ping messages to the EMS every 5 or 30 seconds.

There is a lot of information available from the GUI screen. It displays the manager's telephone number and mobile telephone number, the result of the status, which are office, IP, type of NE, maker, status of EMS and test time of each EMS test, and the history of the status of the EMS test. There is also a special function for sending an SMS (short message system) to the manager.

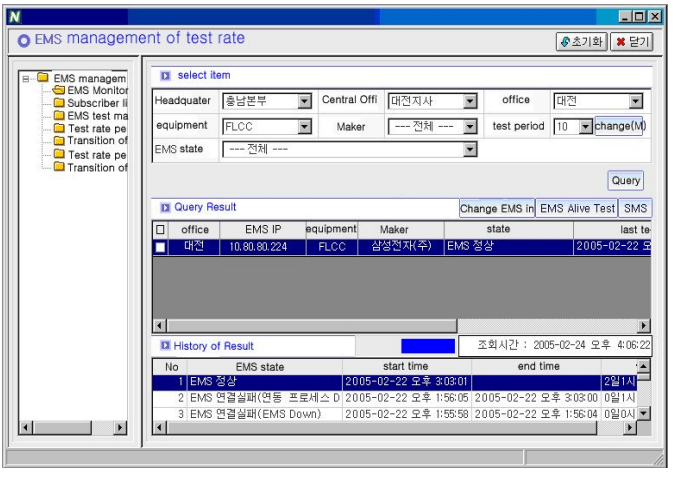


Fig. 5. Screen dump of EMS Monitoring

4.2 Customer's Subscriber Line History Management

This displays the history of the instant test and the detailed test, the information displayed screen order by office, service, type of NE and maker. Furthermore, it can perform an instant test for the specific customer's number.

The operator wants to know the history of each NE and the method for solving the problem in order to prevent the same mistake in managing the EMS or to analyze the status result of the EMS. And he wants to level up the test success rate using this function.

METF also manages the test result data of each EMS sorted by headquarters, central office, office, service, EMS and maker. One is the test rate per office, which displays statistics on the office's test success rate and the chart to understand the test rate transition of the office. The other is the test rate per EMS, which displays statistics about each EMS's test rate and the chart to understand the test rate transition of each EMS. And the operator can ask for the result of each testing history and save an Excel file when he wants to save this content to a file.

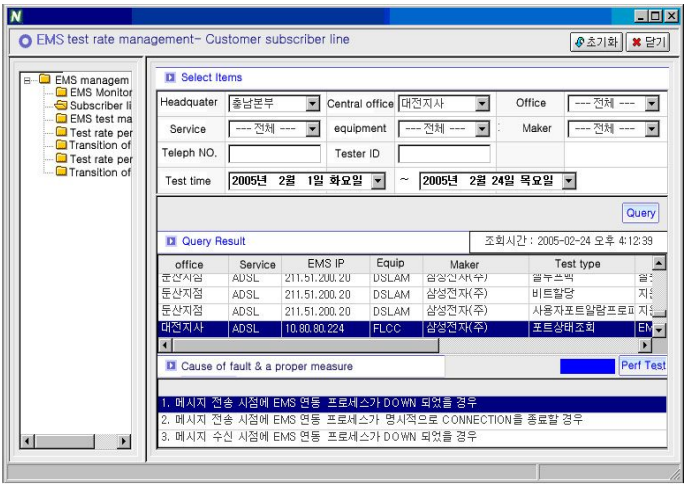


Fig. 6. Screen dump of the customer's subscriber line history management

5 Proactive Maintenance

The NeOSS-ADM-TM can provide proactive maintenance services before customers can report their complaints. Operators can be notified of planned maintenance scheduling by using the function of service-affected: customer identification, performance/fault trend check, and performance/fault analysis. When the Customer Center operator receives a customer's complaint, the NeOSS-SA operator can be notified of the status of the access network without testing.

The Network Management function of the NeOSS-ADM-TM consists of fault, performance and configuration functions. Fault function gathers event/fault data from the access network, which consists of DSLAM, FLC-A/B/C/D, AGW, IDLC-MUX and Exchange. Performance functions, which have the function of performance collection, gather performance data. The NeOSS-ADM-TM retrieves the configuration data of the access network, which is saved in the database of the NeOSS-FM (Facility Management).

The NeOSS-ADM-TM provides the proactive test function. Even without receiving the customer's complaint, office operators can generally register the subscriber line (telephone number), the range of cable/pair, the important customers, the step of trouble processing, and the facility (FLC-A/B/C/D, AGW, IDLC-MUX and/or DSLAM) in order to test during nighttime, when there are few customer complaints by telephone. Thus, we can perform metallic line tests or high-speed internet service line performance tests every nighttime.

If operators want to change the registered record, they can ask and modify the registered record. The next day, the operator asks and analyzes the test result, and in case of a fault result, the operator turns over this customer to trouble processing. In

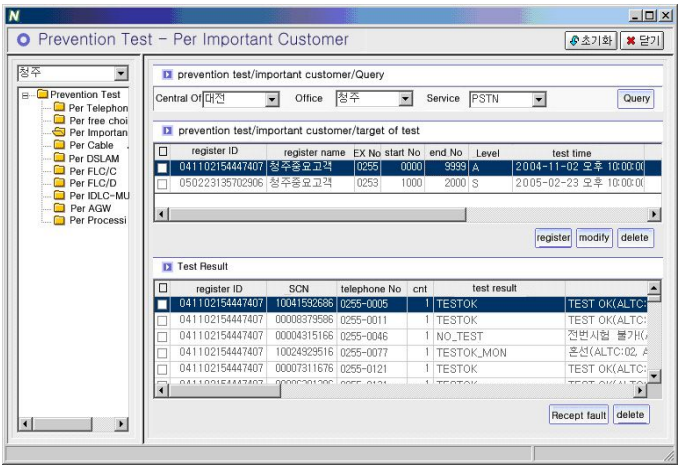


Fig. 7. Screen dump of prevent testing

the stage of trouble processing, the operator executes the detailed test and dispatches the technician.

5 Conclusion

The major features and functions of the NeOSS-ADM-TM system, which provides NE maintenance for the maintenance of the normal status of NE's EMS and proactive maintenance for repairing customer faults, are described at this paper.

The NeOSS-ADM-TM performs the test for each NE and subscriber line using the TU (Testing Unit) or inter-operation with the NE's EMS. Then, the operator requests for the test result and they compare the normal condition. If the operator concludes that the test result is fault, the operator turns over this customer to trouble processing. In the stage of trouble processing, the operator executes the detailed test and dispatches the technician.

Nowadays, customer requirements for high-level quality PSTN or ADSL services are more varied, and there are a lot of new and varied telecommunication network devices. Many other services and new facilities will also appear in the near future, so we have to adapt to that. Therefore, KT has designed the NeOSS-ADM, which integrates the access domain system of service provision, service management and network management.

The developed NeOSS-ADM-TM improves customers' trust in our company, KT, thus giving us the competitive power over any other company's service quality. The NeOSS-ADM-TM also supplies the basis for Before Service for all customers.

There is a need to further study the security issues for the NeOSS-ADM-TM and promote the test success rate & test correctness rate. We are preparing new algorithms for test correctness rate and will deploy a new version of the NeOSS-ADM-TM later this year.

References

1. Shin-ho Choi: Building a Service Assurance System in KT, NOMS, (2004)
2. Bok-Gyu Hwang: Integrated Network Management System for Access Network in KT, APNOMS, June (2003)
3. Mae-Hwa Park**:** The NeOSS-ADM development, Korea Telecom Technical Review, Vol.17, No 4, Dec (2003)
4. TMForum: New Generation Operations Systems and Software (NGOSS) Architecture, March (2003)
5. TMForum: Enhanced Telecom Operations Map (eTOM®): The Business Process Framework-for the Information and Communications Services Industry, June (2002)
6. Hyunmin Lim**:** Web-based Operation Support System for the maintenance of Access Networks in Korea Telecom, NOC2000, June (2000)

Self-management System Based on Self-healing Mechanism*

Jeongmin Park, Giljong Yoo, Chulho Jeong, and Eunseok Lee**

School of Information and Communication Engineering Sungkyunkwan University
300 Chunchun Jangahn Suwon, 400-746, Korea
{jmpark, gjyoo, chjeong, eslee}@ece.skku.ac.kr

Abstract. Systems designed to be self-healing are able to heal themselves at runtime in response to changing environmental or operational circumstances. Thus, the goal is to avoid catastrophic failure through prompt execution of remedial actions. This paper proposes a self-healing mechanism that monitors, diagnoses and heals its own internal problems using self-awareness as contextual information. The self-management system that encapsulates the self-healing mechanism related to reliability improvement addresses: *(1) Monitoring layer, (2) Diagnosis & Decision Layer, and (3) Adaptation Layer*, in order to perform self-diagnosis and self-healing. To confirm the effectiveness of self-healing mechanism, practical experiments are conducted with a prototype system.

1 Introduction

Self-management entails many different aspects, resulting in distinct dimensions of control. For instance, IBM's autonomic computing initiative views self-managing systems as typically exhibiting a subset of four capabilities: self-configuring (adapt automatically to dynamically changing environments), self-healing (discover, diagnose, and react to disruptions), self-optimizing (monitor and tune resources automatically), and self-protecting (anticipate, detect, identify, and protect themselves from any attacks) [1].

As the complexity of distributed computing systems increases, systems management tasks require significantly higher levels of automation. Thus, there is a growing need for experts who can assure the efficient management of various computer systems. However, management operations involving human intervention have clear limits in terms of their cost effectiveness and the availability of human resources [1]. Of all computer problems, about 40% are attributable to errors made by the system administrators [2]. Thus, the current system management method, which depends mainly on professional managers, needs to be improved.

To solve these problems when they do arise, there is a definite need for effective self-healing system. The existing self-healing systems consist of a 5-step process,

* This work was supported in parts by Ubiquitous Autonomic Computing and Network Project, 21th Century Frontier R&D Program, MIC, Korea under ITRC IITA-2005-(C1090-0501-0019), and grant No, R01-2006-000-10954-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 372 – 382, 2006.
© Springer-Verlag Berlin Heidelberg 2006

including *Monitoring, Translation, Analysis, Diagnosis and Feedback*. This architecture has various drawbacks are presented as follows.

- Because the existing system is a log-based system, if an error or problem arising in a component does not generate a log event, it can't heal the problem or error.
- Increased log file sizes and frequency.
- The wastage of resources (such as 'RAM', 'CPU', etc).
- A lot of dependency on the administrator and vendor.

Consequently, this paper proposes ***Self-Management System based on Self-healing Mechanism*** which incorporates several functions designed to resolve the above mentioned problems, namely i) the minimization of the resources required through the use of a single process (Monitoring Agent), ii) the use of a Rule Model which offers different appropriate adaptation policy according to the system situation, viz. '*Emergency'*, '*Alert'*, *'Error' and 'Warn'*. iii) For the sake of rapid and efficient self-healing, we use a 6-step process. The proposed system is designed and implemented in the form of a prototype, in order to prove its effectiveness through experimentation.

The paper is organized as follows: in Section 2, we summarize related works. In Section 3, we describe the proposed system architecture, and in Section 4, we discuss its implementation and evaluation. Finally, in Section 5, we present our conclusions

2 Related Works

Oreizy et. al. [4] proposed the following processes for self-adaptive software: Monitoring the system, Planning the changes, Deploying the change descriptions and Enacting the changes [5][6][7][8]. The Adaptive Service Framework (ASF) [10] proposed by IBM and CISCO consists of a 5-step process, including *Monitoring, Translation, Analysis, Diagnosis and Feedback*. These 5 processes are applied in the form of self-adaptive behaviors. The functions of the ASF are as follows: firstly, the *Adapters* monitor the logs from the various components (*Monitoring*). Secondly, the *Adapter* [9] translates the log generated by the component into the CBE (Common Based Event) format (*Translation*). Thirdly, the *Autonomic Manager* [9] analyzes the CBE log. This step identifies the relationship between the components through their dependency (*Analysis*). Fourthly, the *Autonomic Manager* [9] finds the appropriate healing method by means of the *Symptom Rule* [9] and *Policy Engine* [9] and then applies the healing method to the applicable component. The feedback from the *Resource Manager* [9] enables the system to heal itself (*Diagnosis and Feedback*). Finally, in the event that the component has a critical problem or one which cannot be solved easily, the Autonomic Manager sends a *Call Home Format*[1] message to the Support Service Provider (SSP)/Vendor, requesting them to find a solution.

However, the problems in these existing systems can be summarized as follow:

The size of the log in the CBE format is larger than that of the untranslated log. (This drawback will reduce the system performance)

[1] Call Home Format: This is the message transmission code between healing system and SSP/Vendor that IBM&CISCO are undertaken for standardization. http://www.cisco.com/application/pdf/en/us/guest/partners/partners/c644/ccmigration_09186a0080202dc7.pdf

- The disk, CPU and memory usage drastically increase in the process of conversion, due to the complex calculations involved.
- The ASF has as many Adapters [9, 10] as there are components, and this may cause a problem of insufficient resources, particularly in the case of handheld devices used in ubiquitous environments.
- The ASF requires a high healing time, because immediate action time corresponding to emergency situation is shortage.
- Furthermore, in the event that the component does not generate the log, it is impossible for the system to heal itself.

3 Proposed System

The adaptation of proposed system is divided into three layers, such as Monitoring Layer, Diagnosis & Decision Layer and Adaptation Layer. Fig. 1 shows an architecture for self-healing, which is composed of the Monitoring Agent, Diagnosis Agent, Decision Agent and Searching Agent. The proposed system consists of 6 consecutive processes, viz. Monitoring, Filtering, Translation, Analysis, Diagnosis and Decision.

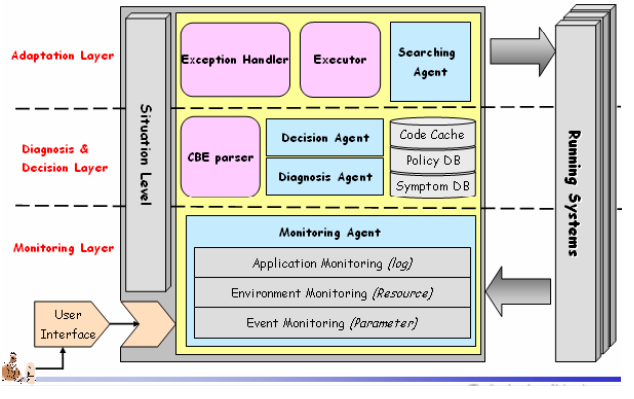


Fig. 1. Architecture of self-management system

3.1 Monitoring Agent (*Monitoring*)

As shown in Fig.2, the functions of the Monitoring Agent are as follows:

- It monitors resource (such as RAM, CPU, etc) status and the size of the log file generated by the components.
- To deal with errors or problems arising in the component that do not generate log events, it monitors error events arising in the operating system, in order to detect problems or errors concerning these components.

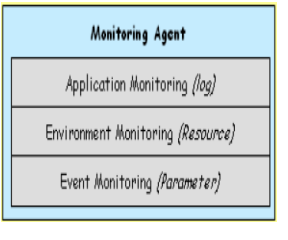


Fig. 2. Monitoring Agent's behavior

Through resource status, log files and error events, if the Monitoring Agent detects suspicious events of the components, it executes the CBE Parser in the Diagnosis & Decision layer.

3.2 CBE Parser (Filtering and Translation)

The functions of the CBE Parser are as follows:

- It gathers the information provided by the Monitoring Agent.
- It filters error context from normal log context File. The error context is filtered by means of designated keywords, such as *"not", "error", "reject", "notify"*, etc. It translates the filtered error context into the CBE format and deliveries the translated information to the Diagnosis Agent

3.3 Diagnosis Agent (Analysis and Diagnosis)

The first major attribute of a self-healing system is self-diagnosing [9]. The Diagnosis Agent analyzes the CBE log, resource information (received from the Monitoring Agent) and the dependency of the components, and then diagnoses the current problem (through *the Symptom DB*). It provides technology to automatically diagnose problems from observed symptoms. The results of the diagnosis can be used to trigger automated reaction. Using the loging service existing in the operating system, as shown in Table1, it classifies the Error Event, and sets up the priorities. The results of the diagnosis recognize the situation level.

Error Level	Priority
Emergency	1
Alert	2
Error	3
Warn	4

Fig. 3. Classification of the error event

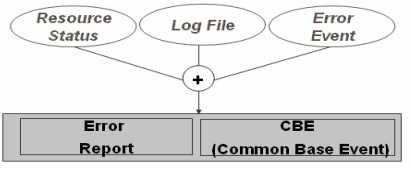


Fig. 4. Diagnosis Agent's behavior

```
Context Predicate
   Identify (MainObject, Object_Topology)
   Situation (Error_Topology)
   State (Resource Type, Usage)

   MainObject: Apache, Oracle, Websphere, Linux, Windows …
   Object_Topology: OS, WAS, DBMS …
   Error_Topology: Emergency, Critical, Elert, Error, Warn…
   Type

Example of Context Predicate
   Identify (Apache, WAS)
   Situation (Emergency)
   State (CPU, 80)
   State (RAM, 70)
   State (Max_Client, 70)

   Identify (Apache, WAS) ∧ State (CPU, 80) ∧ State (RAM, 70)
         ∧ State (Max_Client, 70) -> Situation(Emergency)
     ⇒ Policy 1 (Apache, Control Max_ClientNum)
     ⇒ Policy 2 (Apache, Kill NotUsedProcess)
     ⇒ Policy 3 (Apache, Restart)
```

Fig. 5. Reconfiguration of System level

As shown in Fig 4, the Diagnosis Agent generates the Error Report and modifies the CBE. The Error Report is an administrator document, and the CBE is a document for the system. The following is the algorithm performed by the Diagnosis Agent Using the first-order logic, we can recognize the situation level of system and represent the policy for it. Fig. 5 illustrates context predicate and its example.

3.4 Decision Agent and Executor (*Decision and Execution*)

The log contexts generated by the component as result of observing an error is referred to as an error report. The Decision Agent can take proactive and immediate action corresponding to Emergency situation (Priority '1') through the code cache. The Decision Agent consists of the CBE Log Receiver, Resource Collector, Adaptation Module and Executor, as shown in Fig 6. When the Decision Agent receives the CBE log from the CBE Log Receiver, the Resource Collector gathers the CPU information, Memory information, process information and Job Schedule information, in order to deliver it to the Adaptation Module. The Adaptation Module is in possession of the threshold values pertaining to the gathered resource information. According to the threshold value, a suitable policy is implemented. The Executor then executes the best healing method. The Decision Agent handles emergency situations in accordance with the Rule Model, and applies the best healing method.

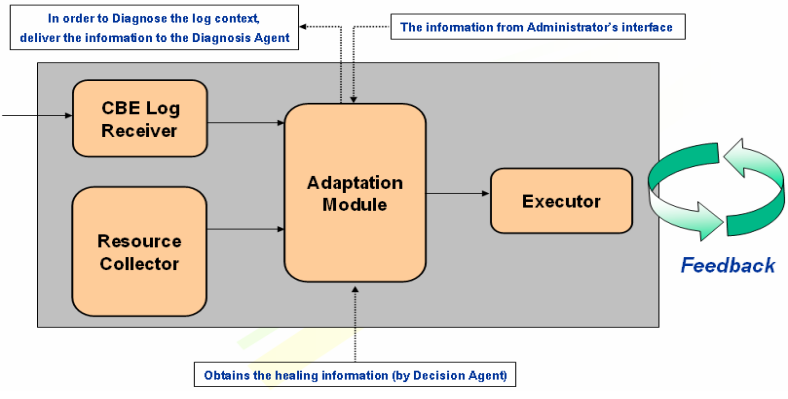


Fig. 6. Architecture of decision Agent

Through the information delivered by the Diagnosis Agent, The Decision Agent determines the appropriate healing method with the help of the *Policy DB*. It also receives feedback information from the administrator in order to apply the more efficient healing method. The Information received from the Diagnosis Agent is used to determine the healing method. The Decision Agent determines the solutions that can be classified into root healing, temporary healing, first temporary healing and second root healing. Temporary healing is a way of resolving a problem temporarily, such as disconnecting a network connection, assigning temporary memory. The root

healing is the fundamental solutions on the diagnosed result, including re-setting, restarting, and rebooting. The Decision Agent stores the methods in the DB as below Table 1, and decides how to select the appropriate healing method.

Table 1. Decision Table

PROBLEM	TEMPRARY SOULTION	ROOT SOLUTION	CURRENT JOB	FUTURE JOB	AVAILABLE MEMORY	DECISION	FEEDBACK
CONNECT	CHANGE CONFIGURATION	REINSTALL	NULL	NULL	80%	T	POSITIVE
AVAILAVBLE	NULL	RESTART	NULL	NULL	80%	R	POSITIVE
OVERFLOW	ALLOCATE MEMORY	REBOOT	BACKUP	NULL	30%	TR	POSITIVE

The Table is the table to determine the optimal resolution method by analyzing given attributes. Looking at the **DECISION** column, when placed under the current diverse context, it helps to determine **R** (Root Solution), **T** (Temporary Solution), or **TR** (first Temporary Solution, second Root Solution). The **FEEDBACK** Column is showing feedbacks that were executed by the System Agent to heal the system. The Decision Agent compares the fields with the information received by the System Agent, these fields are **CURRENT JOB**, **FUTURE JOB** and **AVAILABLE MEMORY**. If the value of the **FEEDBACK** Column is **POSITIVE**, the appropriate method is determined.

3.5 Searching Agent

It is used to search the vendor's website for the knowledge required to solve the problem. This Agent uses search engine (such as *Google*). It sends the resulting search information to the administrator.

3.6 Code Cache

The Code Cache is used to provide healing code to solve the error of the component arising in the *emergency situations*.

3.7 Rule Model for Self-healing

Separation of concerns [14] often provides some powerful guidance to complicated problems. Separation of concerns has led to the birth of many novel technologies such as aspect-oriented programming [15], subject-oriented programming. We used Rule Model approach [16] for self-healing: extract scattered rules from different procedures. Events are what should be monitored and handled. We consider only events that are involved in adaptive rules: Errors or failures that occur when procedures are executed. Most modern languages (such as Java) provide exception capture mechanisms. These types of events need self-healing to make software robust. Using the *Rule Model* that reconfigures services of the proposed system, agents can apply the appropriate adaptation policy. Fig 7 shows that suitable actions are selected via a Rule Model.



Fig. 7. Rule Model of the proposed system

The Rule Model document identifies the different events that can be applied, namely the "*Emergency*", "*Alert*", "*Error*" and "*Warn*" *situation*. Theses situations have actions, linked to their respective situation, and then services of the proposed system are reconfigured by this strategy. We can understand their behavior from the above document. If the agent classifies the current situation as an emergency situation, it acts transformed code to heal the component. The above rule represented in XML is transformed to:

```
if(Scope=="Application"){
    if(ComponentName=="Apache Web Server 2.0.52"){
        if(TypeKind=="File"){
            if(Apache.MaxClientNum()>=140)&&(Net.Latency()>=2)
                Apache.MaxClientNum()=180;
        }
    }
}
else if(Scope=="System"){
    if ((ComponentName=="Apache Web Server 2.0.52"){
        if(TypeKind=="Process"){
            if(CPU.Utilization()>0.7&&RAM.Utilization()>=0.8)
                MySystem.KillNotUsedProcess();
        }
    }
}...
```

4 Implementation and Evaluation

The Fig. 8 is the sequence diagram of the messages is exchanged among self-healing agents. Each of the agents interacts with each other. As show in Fig 1, according to the architecture of the proposed system, we represented the behavior of the agents. The implementation environment is as follows: we employed JAVA SDK1.5, and used Oracle9i as the DBMS. Also we used JADE1.3 for the Agent development.

The sample log used for the self-healing process was the contexts of log that are generated with *APACHE*. We implemented *the Agents* of the proposed system (in the form of a JADE Agent Platform [13]). Each of the agents is registered with each of

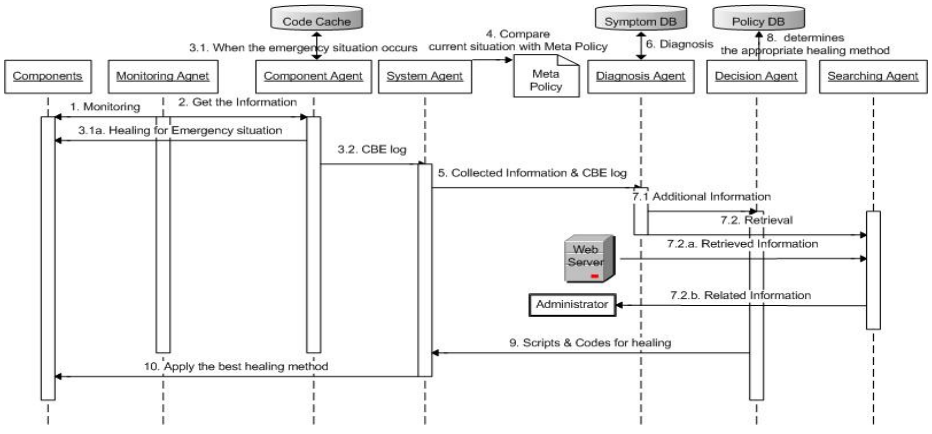


Fig. 8. Message Sequence Diagram

the containers in JADE Agent Platform, and the ACL (Agent Communication Language) is used to communicate among the agents. We performed the simulation using six agents.

The Fig 9 shows extracted log data and resource monitoring for self-management

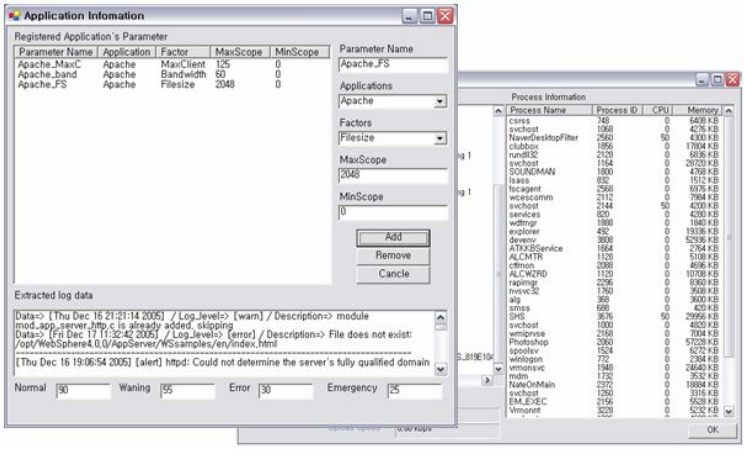


Fig. 9. The result of the Monitoring Agent

The proposed system was evaluated and compared qualitatively and quantitatively in terms of the Log Monitoring Module, the Filtering & Translating Efficiency, and the healing Time. **(a) Log Monitoring Test.** In the existing system, if the number of components is Ω, the system has to have Ω processes to monitor the log. In the proposed system, however, only one process is needed to monitor the log, as shown in Fig. 10. In this figure, the proposed system demonstrates its ability to stay at a certain level of memory usage, even when the number of components is increased.

(b) Filtering & Translation Efficiency Test. In the proposed system, the Component Agent searches for a designated keyword (such as “not”, “reject”, “fail”, “error”, etc.) in the log generated by the components. By using this approach, we were able to increase the efficiency of the system, in terms of the size of the log and the number of logs. We analyzed up to 500 logs, filtered out those logs not requiring any action to be taken, and evaluated the number and size of the logs in the case of both the existing and proposed systems. As a result of the filtering process, only about 20% of the logs were required for the healing process, as shown in Fig. 10. Therefore, the proposed system reduces the number and size of the logs, which require conversion to the CBE format.

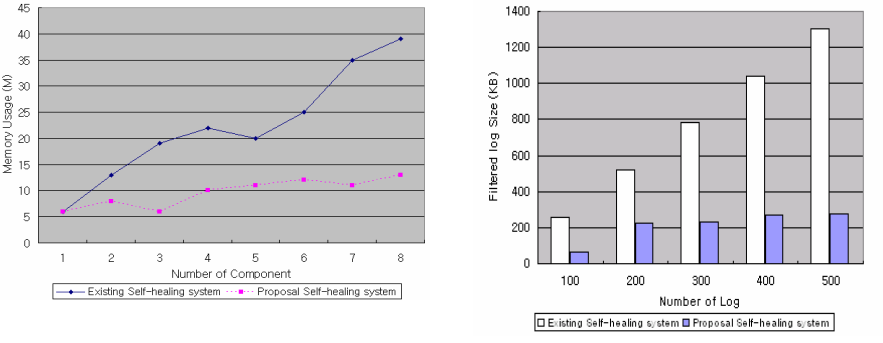


Fig. 10. Memory usage and comparison of size and number of logs

(c) Average Healing Time Measurement. We measured the Average Adaptation Time arising in the existing self-healing system and the proposed self-healing system. The detail evaluation was included below.

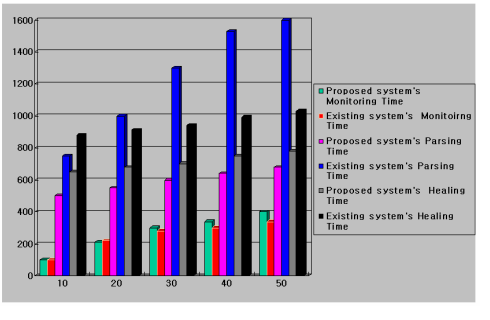


Fig. 11. Comparison of Adaptation time

For each adaptation time, as shown in Fig.11, we verified that the proposed system’s parsing time and healing time are fastest than the existing system’s, and rapidly responded problems arising in the urgent situation. However, because the number of monitoring factors is a little more, although the proposed system’s monitoring time

was relatively a little load through providing the meaningful much more information we verified that high quality of healing have been increased by monitoring information. Although In the event that the error component does not generate log, we couldn't measure the healing time arising in the existing self-healing system because the existing system was log-based healing system.

5 Conclusion

This paper has described self-management system for reliable system. The monitoring layer consists of modules for monitoring the information such as log context, resource, configuration parameters. Once monitoring module in the monitoring layer detects an anomalous behavior and presumes that the behavior needs to be treated. In the abnormal phase, modules in the diagnosis & decision layer are triggered. The diagnosis & decision layer constitutes modules that filters, translates, analyzes, diagnoses the problems, and decides its strategy. Finally, the adaptation layer composes modules that execute the strategy selected in diagnosis & decision layer.

The advantages of this system are as follows. First, when prompt is required, the system can make an immediate decision and respond right away. Second, the Monitoring module monitors the generation of the log on the fly, thus improving the memory usage. Third, before converting the log into the CBE (Common Base Event) format, filtering is performed in order to minimize the memory and disk space used in the conversion of the log. Fourth, it provides the fast healing time. Fifth, using the *Rule Model*, the appropriate adaptation policy is selected. However, further decision mechanism is likely to need to select the appropriate adaptation policy. Moreover this approach may be extended for the intellignent sub-modules in the Diangosis & Decision layer.

References

1. http://www.ibm.com/autonomic
2. IBM: Autonomic Computing: IBM's Perspective on the State of Information Technology, http://www.ibm.com/industries/government/doc/content/resource/thought/278606109.html.
3. Jeffrey O. Kephart David M. Chess IBM Thomas J. Watson Research Center: The Vision of Autonomic Computing, IEEE Computer Society, January (2003)
4. P. Oreizy et. al.: An Architecture-Based Approach to Self-Adaptive Software, IEEE Intelligent Systems, Vol. 14, No. 3, May/June (1999) 54-62.
5. Garlan, D. and Schmerl, B.: Model-based Adaptation for Self-Healing Systems, Proceedings of the First ACM SIGSOFT Workshop on Self-Healing Systems (WOSS), South Carolina, November (2002) 27-32.
6. G. D. Abowd, R. Allen and D. Garlan.: Formalizing style to understand descriptions of software architecture, ACM Transactions on Software Engineering and Methodology, Vol. 4, No. 4, October (1995) 319-364.
7. D. Batory and S. O'Malley: The Design and Implementation of Hierarchical Software Systems with Reusable Components, ACM Transactions on Software Engineering and Methodology, Vol. 1, No. 4, October (1992) 355-398.

8. M. Bernardo, P. Ciancarni and L. Donatiello: On the formalization of architectural types with process algebras, Proceedings of the 8th International Symposium on Foundations of Software Engineering, November (2000) 140-148
9. B. Topol, D. Ogle, D. Pierson, J. Thoensen, J. Sweitzer, M. Chow, M. A. Hoffmann, P. Durham, R. Telford, S. Sheth, T. Studwell: Automating problem determination: A first step toward self-healing computing system, IBM white paper, October (2003)
10. J. Baekelmans, P. Brittenham, T.Deckers, C.DeLaet, E.Merenda, BA. Miller, D.Ogle, B.Rajaraman, K.Sinclair, J. Sweitzer: Adaptive Services Framework CISCO white paper, October (2003)
11. Hillman, J. and Warren, I. Meta-adaptation in Autonomic systems, In Proceedings of the 10th International Workshop on Future Trends in Distributed Computer Systems (FTDCS), Sozhou, China, May 26-28 2004
12. David Garlan, Shang-Wen Cheng, Bradley Schmerl: Increasing System Dependability through Architecture-based Self-repair, Appears in Architecting Dependable Systems, de Lemos, Gacek, Romanovsky (eds) 2003, © Springer-Verlag.
13. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB, formerly CSELT) Giovanni Rimassa (University of Parma): JADE PROGRAMMER'S GUIDE
14. David. Parnas, Designing Software for Extension and Contraction, 3rd International Conference on Software Engineering, pp. 264-277, 1978.
15. Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin, Aspect-Oriented Programming, In proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finland. Springer-Verlag LNCS 1241. June 1997.
16. Qianxiang Wang, Towards a Rule Model for Self-adaptive Software ACM SIGSOFT Software Engineering Notes Page 1 January 2005 Volume 30 Number 1 pp. 1-5.

Experiences in End-to-End Performance Monitoring on KOREN*

Wang-Cheol Song[1,**] and Deok-Jae Choi[2]

[1] Department of Computer Engineering, Cheju National University, Jeju 690-756, Korea
philo@cheju.ac.kr
[2] Department of Computer Engineering, Chonnam National University, Gwangju 500-757, Korea
dchoi@chonnam.ac.kr

Abstract. As the network technology has been developed, the Next Generation Internet (NGI) such as Internet2, KOREN, KREONET2 and etc has been deployed to support bandwidth of Giga bps. And, various applications such as the video conference, the tele-surgery and etc that require high bandwidth has been developed and operating on the NGI, especially KOREN and KREONET2 in Korea. When the applications are operating and happen to face some problems, various tools for traffic measurement are usually used to analyze them, and provide traffic information in each link on total flow path or other metrics such as end-to-end delay, jitter and etc in most cases. However, although with these tools most of network problems can be properly analyzed, the problem in the user's view point can not be resolved so that which part of networks and/or which ones among user's systems and nodes on the flow path cause the problems for the user's flow is not discriminated Therefore, this paper designs a end-to-end performance monitoring system for the user's flow that the user can access performance data for user's flow on the flow route, and describes the experience about deployment on KOREN.

1 Introduction

As the network technology has been developed, the Next Generation Research Networks have been deployed to support bandwidth of Giga bps. As the international research network such as GLORIAD[3] as well as the domestic research network such as Internet2 in U.S.A.[1] and KOREN[2] in Korea appear, there have been various collaborations for research and performance through the network. Especially, many activities such as the video conference, the tele-surgery and etc through KOREN and KREONET2 have been internationally performed by using applications that transport the high quality multimedia data of the high-definition (HD) level in real time.

When experiencing these activities to use the applications, various problems are often faced, but usually the end users should realize that they have no special means without discussing the phenomena with the end users on the opposite site. For examples, if you hear noise or watch the distorted video stream from the network, you can

* This work has been supported by NCA, Korea.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 383 – 392, 2006.
© Springer-Verlag Berlin Heidelberg 2006

not discover whether it comes from the network problems or from system itself on either site. Therefore, if information about users' flow can be provided to the users themselves as a way to solve these problems in network events, we think that they can as least understand whether the source of the faced phenomena is network problems or not.

As many open source tools have been already developed to investigate the network situations, we can easily get very various kinds of information from networks, but most of them do not consider the user's individual flow information but just show current traffic amount and available bandwidth on the path or some link. Although much bandwidth is available on the flow path, we have experienced in many cases that users can not use fully the bandwidth if the end systems are not tuned properly. So, enough bandwidth means neither the user's flow can use it nor the user currently uses it. And, the network operator can collect useful information for flows from every router, but as it is provided specially depending upon the user's request, users can not easily access user's individual flow information.

Therefore, this paper designs and deploys the end-to-end performance monitoring system on KOREN, so that users can easily get information about their flow's information in the user's viewpoints and analyze their communication status by themselves. This paper consists of four sections: after introduction, we explain the system architecture in section 2, and describe the deployment on KOREN in section 3. Finally, section 4 concludes and describes the further research.

2 The Architecture of the End-to-End Performance Monitoring System

When the user's flows are transported through the network, *which flow is transported via which routers* can be investigated as follows: (i) each router in the network provides necessary information about the specific flow, (ii) the dynamic route for the flow from the information is extracted by using some tool, and (iii) the users individually get information about the rate and/or the bandwidth used in every router on the route for the flow. For the purpose, we use the RTFM (Real-time Traffic Flow Measurement) Traffic Measurement Architecture [5] as the basic architecture for obtaining the flow information. And, as it has been already developed in IETF, we extend it to design the network-wide model.

RTFM provides a general framework for describing and measuring network traffic flows. Its basic architecture consists of meter, reader and manager, and the analysis application can ask the information to reader as shown in figure 1. Each component is defined as follows [5]:

- MANAGER: an application which configures 'meter' entities and controls 'meter reader' entities.
- METER: Meters are placed at measurement points determined by Network Operations personnel. Each meter selectively records network activity as directed by its configuration settings.
- METER READER: A meter reader transports usage data from meters so that it is available to analysis applications.

- ANALYSIS APPLICATION: An analysis application processes the usage data so as to provide information and reports which are useful for network engineering and management purposes.

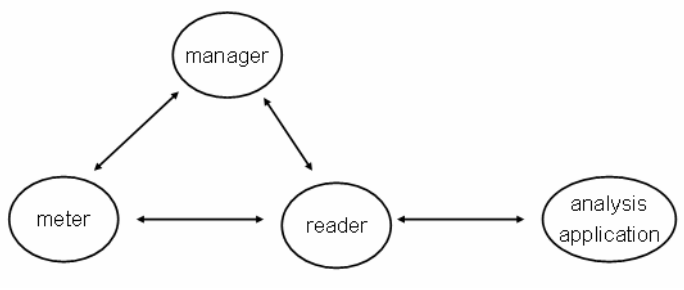


Fig. 1. Architecture of RFC 2722

This architecture has been already taken by NeTraMet [6], and IPFIX [7] also takes the similar approach. With this basic architecture, we intend to design the network-wide system that each meter on routers collects the flow information from its router and transfer it to the centralized reader so that the reader can analyze it and provide useful flow information to the users. Therefore, we use the meter as the original concept and extend the reader to the centralized one that decides/coordinates to get which flow information from which router on the path. The manager must be designed, but as it is related to the router's security issue and it practically belongs to network operator's authority, design of manager is not considered in this paper.

2.1 Meter

Meter is a component that measures flow traffic in a node, and is RTFM itself. Actually, as this component should be operating on a router, it is necessary to be dependent on the router machine. So, it may be reasonable to select a tool that many routers can have on themselves. As Netflow [4] is well-known as one of the best tools for measuring flow information, and can be deployed on the router of many vendors, we select it as the meter in the design.

2.2 Reader

As we state earlier, in this paper we extend the reader concept of RTFM to the centralized network-wide Reader. It should have role to identify the requested flow's dynamic path, request the flow information to the corresponding routers (meters), and provide the client the received flow information from routers. For the purpose, the reader is designed as shown in the figure 2 that consists of the Request Handler, the Manager, the Coordinator, the Netflow data DB, and the Dynamic Flow Path Detection modules. And, the clients in the figure 2 are entities that are now generating the traffic in the network and want to get their own flow information from Reader.

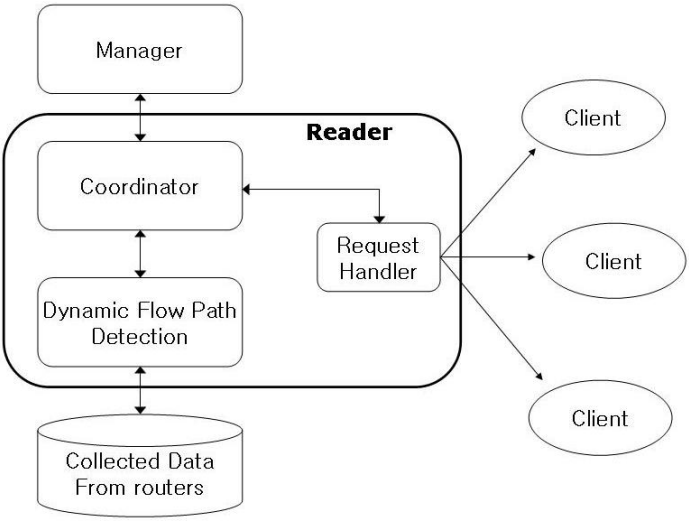


Fig. 2. Reader Architecture

2.2.1 Request Handler Module

The Request Handler has the role that transports the Coordinator the request data from the client and delivers back the result for the corresponding flow data from the Coordinator to the client. The request data from client should contain the source IP/port, the destination IP/port and the periodic monitoring interval. The periodic interval means that the reader is requested to send the flow information to the client repeatedly in the interval. The result data from the Coordinator to the client has the end-to-end flow path, flow rate on each router, and so on.

2.2.2 Manager Module

This module is for configuration of the Reader. This performance monitoring system is to provide the general user's flow information, but it should also have functionality for access control so that privileged users can get the restricted information. As the system chooses the restriction level of information based on the client's IP, the Manager module handles IPs for administration and what kinds of information the system serves. And, the manager configures how much amount of data the system should store. As it is described in the 2.2.4 section, this system actually does not serve long time flow information, and only data within time interval that the manager configures are kept in the DB.

2.2.3 Coordinator Module

The Coordinator calls the modules in the reader in time and coordinates the requests/results to be transferred to proper modules. It mainly calls the Dynamic Flow Path Detection module in every period specified in the request, have the module get the corresponding flow path and flow rate, and transfer it to the Request Handler.

2.2.4 Collected Data

Ideally, this end-to-end monitoring system might be considered to have to have the mechanism to let the reader dynamically find out the flow path upon the client's request, investigate the real-time flow information from the corresponding meters in the path, and provide performance data to the client about the user's flows. However, we must have considered the characteristics and restrictions of the Netflow. The Netflow may be usually restricted to send the flow information to only a single workstation by the network operation center due to some reasons such as router performance, security and etc. Due to the restriction, if you want to share the information from the Netflow with the other measurement tools, first you should prepare some media to store it. Therefore, the end-to-end performance monitoring system in the paper uses the information stored in DB, not collect dynamically and directly information from routers. As the information is not directly collected from meters, the flow data with the full option are transferred and stored from each router to the DB. It means that as the stored data can not reflect the user's request, they should be all of what the meter can generate.

This reader system uses Netflow2Mysql application [8] – an open source code to store the Netflow data in Mysql DB. In figure 3, the left side picture shows the header table to contain the header information of the Netflow data, and right side picture shows the record table to contain the Netflow data themselves.

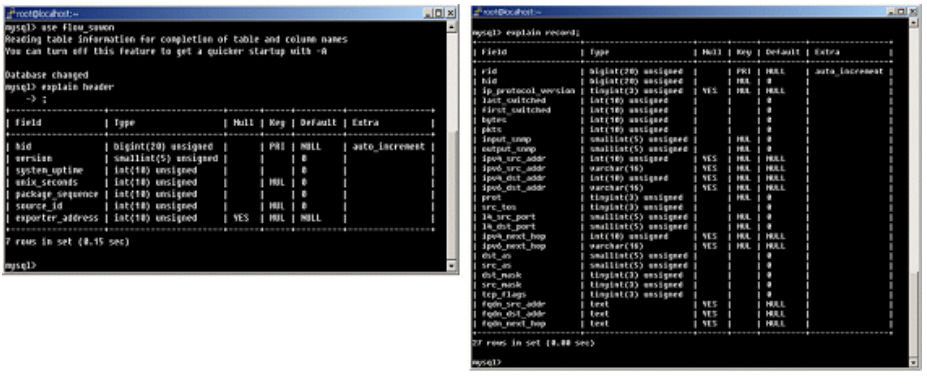
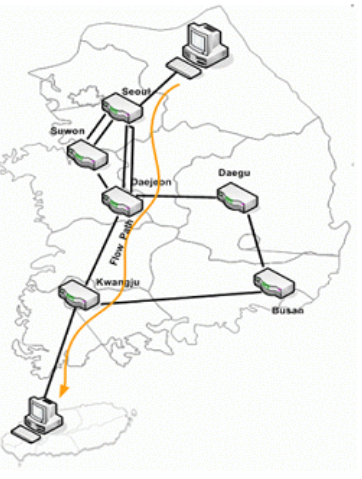


Fig. 3. Netflow data tables stored in Mysql DB

2.2.5 Dynamic Flow Path Detection

This module is for searching the flow path and extracting the flow information in each router on the path from the above DB. This module first extracts the corresponding flow information in the first router on the path by using the tuple {source IP/port, destination IP/port} within the proper duration, and investigates next routers of the flow from the Netflow data. As shown in figure 4, the Netflow data contain the Next hop router IP address field in the record. Matching it with the interface IPs of routers on KOREN provides the flow path *hop-by-hop*. Although we don't request the flow

information directly to the meter in the KOREN routers, this investigation extracts the flow information *hop-by-hop* from the DB. As an example, figure 4 describes that when user systems connected respectively to Seoul node and Gwangju node have video conference by using DVTS[9], the Netflow data collected from Seoul node have the IP address of Daejeon node as the Next hop router IP address value. So, the total flow path can be hop-by-hop searched as Seoul->Daejeon->Gwangju.



Field	Version 5	*Version 5 Catalyst 65k	Version 9	*Version 7 Catalyst 65k
source IP address	Y	Y	Y	Y
destination IP address	Y	Y	Y	Y
source TCP/UDP application port	Y	Y	Y	Y
destination TCP/UDP application port	Y	Y	Y	Y
next hop router IP address	Y	Y 12.1(13)E	Y	Y
input physical interface index	Y	Y	Y	Y
output physical interface index	Y	Y 12.1(13)E	Y	Y
packet count for this flow	Y	Y	Y	Y
byte count for this flow	Y	Y	Y	Y
start of flow timestamp	Y	Y	Y	Y
end of flow timestamp	Y	Y	Y	Y
IP Protocol (for example, TCP=6; UDP=17)	Y	Y	Y	Y
Type of Service (ToS) byte	Y	***PFC3b Only	Y	***PFC3b Only
TCP Flags (cumulative OR of TCP flags)	Y	N	Y	N
source AS number	Y	Y 12.1(13)E	Y	Y 12.1(13)E
destination AS number	Y	Y 12.1(13)E	Y	Y 12.1(13)E

Fig. 4. Dynamic Flow Path Detection

2.3 Client

The Client's functions are very simple. The user who wants to request the flow information fills out about the source IP/port, destination IP/port and update duration items. Then, the Client sends the above items with current time to the Reader and gets the flow information from the Reader repeatedly in the update duration. And, the server application port is well known in most application, but as the client application's port is the ephemeral port number, if a port number is not filled in the request, the Client investigates the port number by using the given IP and port numbers from client system's OS. Figure 5 shows the Client GUI that replies the flow information in text.

We have developed the client as the client/server model, not the web-based model. It is because we consider development of the visualization part in the future. If the system is developed in the web-based model, the server (the Reader) may have the computing burden for visualizing the data. So, we have thought that it would be better the Client receives the flow information as the numerical data and computes the visualization. The computing burden for visualization shall belong to the client machine.

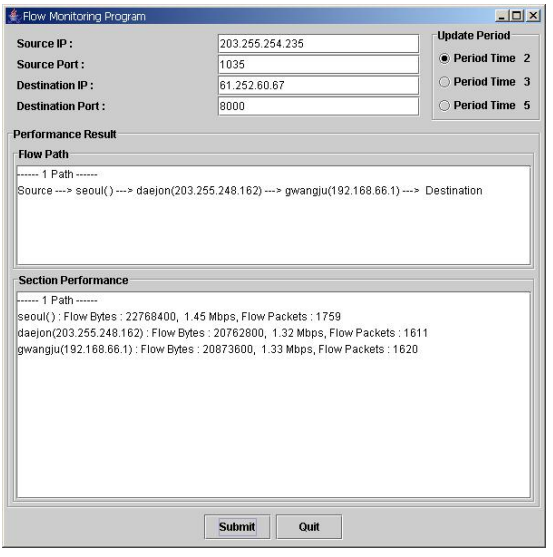


Fig. 5. Client GUI

3 Deployment on KOREN

KOREN is a research network deployed together with KREONET2 in Korea to which only non-profit institutions can have connections, and it has totally 6 nodes such as Seoul, Suwon, Daejeon, Gwangju, Busan and Daegu nodes. Every router on KOREN is running the Netflow to measure the flow information, and supports the SNMP protocol to limited users in some IP addresses. As KOREN NOC (network operation center) supports many tools to analyze the network problems on the web [10], successful deployment of the end-to-end performance monitoring system does not mean that it can exclusively use the Netflow data. Thus, we have located a PC in KOREN NOC for collecting the Netflow data from all of KOREN nodes, and have it store the data as the raw format. We call it the KT Netflow collector.

For our deployment, the Netflow data are fanned-out to the Cheju collector in our laboratory, and the data are stored into the Mysql DB. We use two Mysql DBMS systems for the performance issue as follows: one system is assigned for Seoul, Suwon and Daejeon nodes, and the other system is assigned for Daegu, Gwangju and Busan nodes. Every node uses the Netflow version 5 except that Suwon node uses version 9. On the above environment, we have deployed the end-to-end performance monitoring system in KOREN and any users can easily get their individual flow information and analyze the phenomena when facing some problems. Figure 6 shows that KT Netflow collector gathers the flow information from all of KOREN routers, and fans it out to Cheju collector. And the flow information is appeared in the right side monitor in the figure, and control information for the manager module of the Reader is shown in the left side monitor.

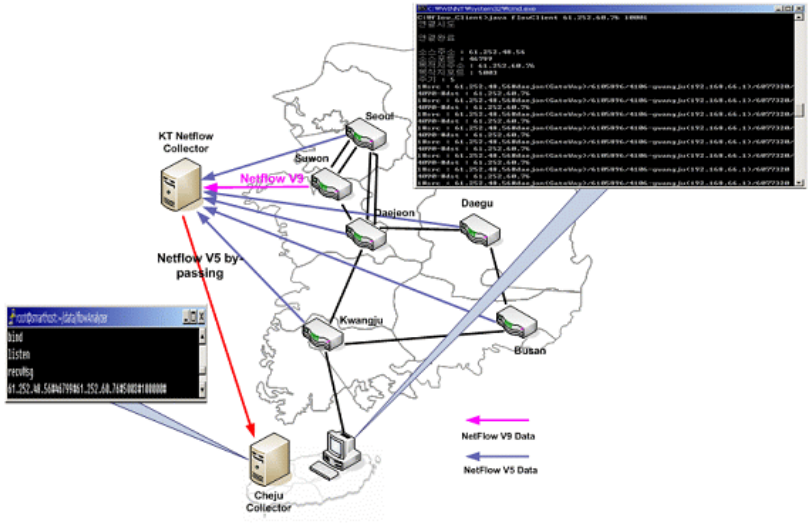


Fig. 6. Deployment in KOREN

3.1 Experiences in Deploying the System

While developing the system, we have faced some limitations in the Netflow although the Netflow has many advantages over the other utilities. First, the Netflow performs 1/10 sampling to generate the flow information in KOREN. Although we have understood that sampling is necessary in order to maintain proper performance of the router in WAN environment, sampling may cause the uncertainty of the flow information to increase. In applications such as DVTS to require high bandwidth, the flow information from Netflow looks still valid, but applications with low data rate can not be detected in the flow information in some cases. Therefore, the system designed in the paper can be used only for flows requiring high bandwidth such as HD-quality multimedia stream.

Second, Netflow can generate its data only in one minute interval at most. It means the minimum interval to generate data is one minutes and it can not timely generate data. It causes synchronization problems among Netflow data from several KOREN routers. If someone want to get his/her flow information at some time, it might be failed because routers can not generate the information timely. Therefore, the flow data provided to users should be at most one-minute earlier data. However, we think this system is still valid because in many cases it may take one or more minutes for users to start monitoring the flow after recognizing the problems, and in some cases users may want to understand only the trends about the traffic.

Third, too many Insert DB commands are required to collect the Netflow data into the DB. At one time, the number of packets received from KOREN was 84 during one minute. The length of each Netflow packet is 1,464 bytes, and the storing unit for a flow in the Netflow data is 48 bytes. Therefore, collecting the Netflow data into DB requires 156,990 Insert commands as follows:

- Data amount received from KOREN nodes during 60 seconds:
  - 84(UDP packets/second from all of KOREN nodes) X 1464(UDP Size) X 60(seconds) = 7,378,560 bytes
- Number of Insert DB command generated for collecting the flow data into Mysql during 60 seconds:
  - 7,378,560 bytes / 47 bytes (bytes per flow) = 156,990 commands

Therefore, although the collected amount of data is not big, too many Insert commands generate the overload of the DBMS, and finally can cause UDP packet drops. This may induce fatal defects that the flow information may be useless. To overcome it, we have adopted the Memory DB and succeeded to watch no packet drops. As this system usually does not need long time flow data, we could accomplish it by using only 2 Gbytes. This solution has been considered in the current status of KOREN. The measured average traffic amount varies link-by-link as follows: 41M bps (in) / 38M bps (out) at Seoul-Taejon link, 760 bps (in) / 310 bps (out) at Taejon-Busan link, 20 Mbps (in) / 20 Mbps (out) at Taejon-Taegu link, 20 Mbps (in) / 26 Mbps (out) at Taejon-Kwangju link, 20 Mbps (in) / 37 Mbps (out) at Taegu-Busan link, and 19 Mbps (in) / 18 Mbps (out) at Busan-Kwangju link. They are available in the KOREN NOC [10].

4 Conclusions

The end-to-end flow monitoring system designed in this paper provides users their own individual flow information in the user's viewpoints. We use the RTFM as the basic architecture and extend it to the network wide system. Our system mainly redesigns the Reader part to investigate the flow path and get the flow information from the corresponding routers on the path. We think this system could be very useful to users who want to use applications requiring high bandwidth on networks. If they face some problems, we think they can easily investigate the cause, analyze it, and request help to the NOC with details about the problems. We have adopted the Netflow as the meter in the system. Due to some limitations of the Netflow we have faced some problems in deployment, but they have been successfully resolved. In the future, we will correlate the flow information supported by this system with traffic information measured by SNMP, and design the visualization effectively to provide very useful information to users.

References

1. http://www.internet2.edu/
2. http://www.koren21.net/
3. http://www.gloriad.org/
4. Cisco System, "NetFlow Services and Applications," White Papers, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm
5. Brownlee, N., Mills, C. and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, October 1999.

6. Brownlee, N., NeTraMet home page, http://www.auckland.ac.nz/net/NeTraMet
7. Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", draft-ietf-ipfix-architecture-07 (work in progress), March 2005.
8. http://public.planetmirror.com/pub/netflow2mysql?vo=24
9. http://www.sfc.wide.ad.jp/DVTS/
10. http://noc.kr.apan.net/

SOA-Based Next Generation OSS Architecture

Young-Wook Woo, Daniel W. Hong, Seong-Il Kim, and Byung-Soo Chang

Network Technology Lab., KT,
463-1 Jeonmin-Dong,Yuseong-Gu,Daejeon 305-811, Korea
{ywwoo, wkhong, sikim, bschang}@kt.co.kr

Abstract. In convergence telecommunication environment, Business Agility plays very important role in the OSS(Operation Support System) when telco provide new merged services to customer on time. But, the OSS also becomes more and more complicated to know even what part of it should be fixed for adopting new services. This paper proposes SOA-based OSS architecture for telecommunication services in order to cope with this situation. We present the designing method of services of SOA and architecture for OSS by investigating the architectural issues of the unit of derived service elements from OSS and designing the most suitable architecture of it. By adopting the represented architecture for OSS, telco can provide new convergence service to customers faster than the competitor on the market.

Keywords: SOA, NGOSS, service, convergence, architecture.

1 Introduction

Recently as new network technologies continue to make appearance in the market, there is a fast growth in converged products as they combine with the existing products. WiBro service will be launched as a Fixed Mobile Convergence product, while Home Networking service where various home information devices converge into a single network and IP TV that allows TV viewing through broadband Internet connection are being introduced.

In line with the launching of such convergence service products, Operation Support System (OSS) must also take on architecture where appropriate management functions can be accommodated at the right time. Still, as the OSS becomes complicated due to the addition of numerous functionalities, it often becomes more difficult to accommodate the management function. In this case, modification of even a single function in the system may cause a reverberating effect on other components, thus, the whole of chain-reaction must be taken into account upon making a modification. This process of adding or modifying a function in the OSS costs a lot of money, not to mention the difficulty in identifying the root cause of a problem when fault occurs after the launch. If legacy components have a high degree of inter-dependency with other components, the reusability of those existing components declines. In case of a convergence service product, due to the fact that it is a combination of several disparate services, it may be much more efficient, in terms of the saving of cost and time, to develop it by utilizing functionalities for the existing services. But if re-utilizing existing components is not easy, provisioning of service at the right time can become a tall task.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 393 – 402, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The most effective solution for this problem may be the Service Oriented Architecture (SOA)[1]. The basic component of SOA, service, can be loosely coupled with each other, thus weakening the relationship between components and implementing OSS by combining modularized services are possible. This way, the system can quickly respond to the addition of management functionalities for the convergence service or new functionalities for existing services. Also, as services become modularized, maintenance will become easy, and with the standardization of service interface, they will have interoperability with other modules that were developed into different platforms[2]. And, the concept of SOA can be a valuable tool for Implementing NGOSS(Next Generation Operation System and Software) program which is suggested by TM Forum.

This paper proposes next generation SOA-based OSS architecture to implement telecom services. After closely examining the architectural aspect of the service unit of the OSS and searching for the most appropriate architecture for processing them, the design methodology and OSS architecture for services, which are the basic unit in SOA configuration is proposed In Section 2, we describe the requirements about NGOSS. And, we introduce the SOA as the OSS architecture which satisfy parts of the requirements for NGOSS. In Section 4, we suggest the method to deduce services from operations from OSS, and suggest next generation OSS architecture by describing constituent part of it. In Section 5, we will represent next generation OSS architecture is applicable to convergence service by giving an example of the WiBro service.

2 Requirements for Next Generation OSS Architecture

The TM Forum which is international group that provides the guidance of network operations and management present the NGOSS(New Generation Operations Systems and Software) as the guide to be the next generation OSS which has more a productive and efficient management architecture than legacy systems. It is designed to deliver measurable improvements in development and software integration environments.

There are key principles that satisfy the NGOSS architecture, such as eTOM[3], SID[4], Contract Defined Interface(CDI), Contract Registration & Trading(CRT), Externalized Process Control(EPC) and Common Communication Vehicle(CCV)[5]. eTOM(Enhanced Telecom Operations Map) is a business framework for categorizing all the business activities that a telecommunication service provider will use. SID(Shared Information and Data model) provides an information and data reference model and common information and data vocabulary from business entity perspective. It is a logically centralized information model which guarantees data integrity and consistency in the OSS. Contract Defined Interface implies that the fundamental unit of inter- and intra-operation should be contracts which is a technology neutral representation of an interface to a service. Contract Registration & Trading implies that NGOSS components can be implemented with NGOSS contracts and the components without knowledge of their physical location in the network. Externalized Process Control is about how independent business processes from the component implementation. It implies operational processes can be changed with

minimum impact on the application. Common Communication Vehicle implies common communication and invocation mechanism is needed for components to communicate with each other in the OSS. These key principles of NGOSS could be implemented by the features of the architectural concept of Service Oriented Architecture

3 Service Oriented Architecture

SOA is a collection of services with appropriate size for users. It is software architecture for developing application completed by linking loosely coupled services as one. Service, which is the basic unit of SOA, refers to a module with a standardized interface independent of a platform. Each service is designed to be loosely coupled with other services to reduce the dependency on other modules. This allows high interoperability as well as easy modification of services. Also, the re-usability of service can be improved and cost reduced due to modularization. Creating a new service by combining existing services becomes easy, and registration and calling of these services regardless of their locations are possible by supporting location transparency. Figure 1 shows the basic Service Oriented Architecture. A service provider implements the service and publishes its description at the Service Registry. A service requester discovers and invokes the desired service registered at the Service Registry for use. Such architecture enables the service requester to search for a needed function from the Service Registry as one can shop for the most appropriate product in the market. Here, the concept of service corresponds to that of contract from NGOSS. Contract Defined Interface can be obtained by using basic architecture of SOA. Contract Registration and Trading which can be obtained by using service registry means that a defined contract shall be available for registration and trading and that location transparency shall be guaranteed.

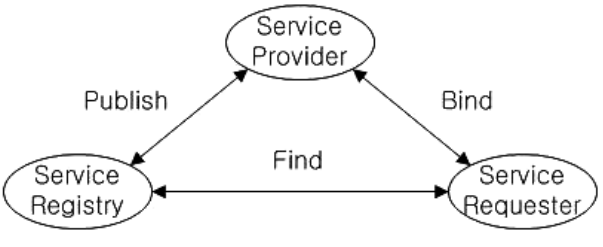


Fig. 1. Basic Service Oriented Architecture

Web service[6] is the leading technology that allows such SOA service implementation. Web service is comprised of standardized technologies independent of platform, such as Universal Description, Discovery and Integration (UDDI)[7], Web Services Description Language (WSDL)[8], and Simple Object Access Protocol (SOAP)[9]. Each of these has the following roles in service implementation: UDDI registers web service and provides mechanism for finding/binding which corresponds to the Service Registry in figure 1; WSDL is used to define the format of the interface

and input/output message for using the web service whose role is to contain the details of services in the Service Registry of figure 1; and SOAP is the SML protocol standard that defines the message format used in web service request and response.

For the implementation of a process that is a coupling of atomic services or composite services, an integrating layer is required. For this, Business Process Management (BPM)[10], Enterprise Service Bus (ESB) and Business Activity Monitoring(BAM) are used. Figure 2 shows a system architecture that includes all of these. BPM is run as a management system and tool for process automation and management as it takes charge of the system implementation with service as the basic unit for business logic implementation. For efficient integration of distributed service components, ESB acts as a middleware that provides process-based integrated support and support for web service-based integrated environment. Aforementioned services are implemented based on consolidated database, and are coupled and orchestrated. Composite Service, which is a collection of orchestrated fundamental services, can also be used as a fundamental service for another round of orchestration. BAM plays a role of monitoring business processes which is developed by BPM and visualized the present condition of business activities of enterprise. Here, BPM and ESB can be considered as means that actually implement the concept of 'Externalized Process Control ' and 'Common Communication Vehicle' of NGOSS.

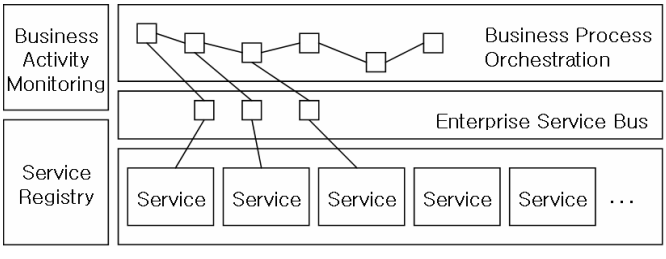


Fig. 2. Service Oriented Architecture with ESB and BPM

4 Realization of SOA-Based Next Generation OSS Architecture

In order to implement SOA-based Next Generation OSS Architecture, service should be defined first. And we need to define the data which will be used when services communicate each other. Before BPM orchestrate service, service management tool is required to manage services. In the following, we describe how services can be derived, a data model, service management, and orchestration in more detail.

4.1 Service Map

To incorporate business logic in a system, the key issue is how to design service, the fundamental unit. Through the analysis of the functional and architectural aspects of services and the review of the performance aspect, the following criteria for decomposing services can be derived.

First, The operation map should be written by decompositioning operations for telecommunication business into hierarchical levels. For example, if Fulfillment is set as level 1 operation, below it at level 2 operation can be lower processes such as Order Entry, Resource Allocation, and Resource Configuration. Each of the Level 2 operation is divided into sub-services, and as the level goes lower, the operation unit is decomposed into the smallest possible unit. For the criteria of such decomposition, eTOM framework of NGOSS can be used as criteria in the service decomposition. And, Service map can be expanded by adding specific telco's own operations.

One of the standards of deriving services from operations may be whether a given operation has dependency on any other operation within the business process. Although an operation may complete its role by definition, there are cases where it is meaningful only as a part of the higher operation. For example, a operation called Issue Service Order does not have any significance on its own, but becomes meaningful only when a different operation in the higher service process closes the service order. Also, a operation may have a dependency on database for its execution. In other words, to run a operation, it may need to fetch data from database, rather than rely entirely on values entered at the operation calling. An example would be a operation called 'Resource Allocation.' This operation automatically allocates optimal resource to the given customer in the fulfillment process. Although customer information is provided when the operation is called, optimal resource information for customer must be searched through resource entities in database. Likewise, if a operation has a dependency on database, implementation of the operation will be affected when the database schema is modified. Therefore, an operation shall be designed to avoid such case and if it is inevitable, it shall be marked so for clear understanding.

To design and manage a system in a holistic way, the criteria of decomposition introduced above can be applied and services can be marked and managed in a table format as shown in table 1. In this paper, this is used as a service map. With such service map, specification and dependencies of operations that constitute the system can be easily discerned, which can be reflected in designing the system. It can be the tool to derive services from the operations of the system.

In the service map, decomposed operations, abstraction units of information, can be denoted by using numbers. The hierarchy of decomposed operations can be denoted in the following way: To denote a level 3 process which belongs to a level 2 process, e.g., '1.3. Resource Config.,' just adds a number to the '1.3.' to make it as '1.3.1 Fixed Resource Config.,' or '1.3.2 Mobile Resource Config.' If a operation is dependent on another, then it can be denoted by adding a prefix 'PD' (from Process Dependent service), and if independent, then 'PI' (for Process Independent service) are added to the numbers for classification. For example, an operation, '1.3.PI Resource Config.,' in table 1 shows that the operation can be used independently When designing a system, this operation can be called by other operations even outside the level 1 operation Fulfillment, disregarding its relationship with other operations. If an operation is dependent on a database, a prefix 'DD' (from Database Dependent operation'), and if independent, 'DI' (from Database Independent operation) are added to the numbers respectively for classification. Operations that are independent from other operations or database are labeled 'PI.DI' and have no

Table 1. An Example of the part of Service Map

Level 1	Level 2	Level 3	Level 4	…
1.PI.DD Fulfillment	1.1.PD.DI Order	1.1.1. PD.DI Issue Service Order	…	…
		1.1.2.PD.DI Order Receipt	…	…
		1.1.3.PD.DI Order Analysis	…	…
	1.2.PD.DD Resource Allocation	…		…
	1.3.PI.DI Resource Configuration	1.3.1.PI.DI Fixed Resource Configuration	…	…
		1.3.2.PI.DI Mobile Resource Configuration	...	…
…	…	…	…	…

problem registering as service to the service registry. But if there is a dependency on either of operations or database, the registration as service to service registry will be decided according to the policy.

When an operation is decomposed as demonstrated above, the overall architecture of a system can be figured out from both business and system aspects when designing. However, operation levels may go excessively deep. If the operation level depth is excessive, the frequency of operation calling will increase, and thus make troubles regarding its performance.

After deriving the service map, services should be derived from the operations. Here, considering time delay is needed. When service is called, there exists time delay for finding the location of service from the service registry and binding the service and parsing the SOAP header. Bad network condition, lack of bandwidth also may cause time delay. Therefore, how many services can be orchestrated in the components without time performance problem should be considered when specific level of operation is decided to be service. Decomposed operations are orchestrated by BPM as shown figure 3.

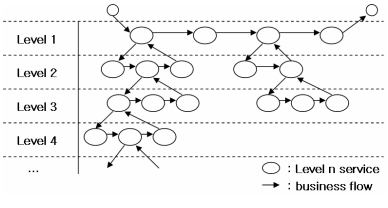


Fig. 3. Business flow of services

If service binding time ‘b’, represents the total time it takes when a service calls to locate another service, binds it and parses the SOAP header, and if Level n service is designed to call Level n+1 services m times, the total time it takes can be calculated as follows:

$$T_{level(n)} = b \times \sum_{x=1}^{m} (T_{level(n+1)}) \quad (1)$$

If Level n service is composed of a number of Level n+1 services, the time it takes to execute Level n service will increase as the number of service decomposition levels increases. Figure 4 illustrates this in a line chart. If the number of service decomposition level is increased, the frequency of recurrent calling of sub-level services will increase, dropping the performance. Therefore, in the implementation step, it has to be decided that up to which level of decomposition shall be implemented as a service unit. The decision can be made based on the number of services, time required to call a service, and processing time of each service.

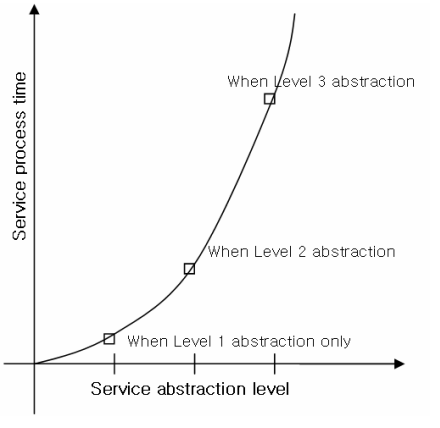


Fig. 4. Conceptual service execution time over service decomposition level

4.2 Consolidated Database

After every operation and service of the system are defined, the data model and repository is needed to contain the communicated data among them. The data can be modeled by analyzing operations from the service map and deriving entities and attributes from them. NGOSS suggests SID(shared Information and Data model) as data model for the common repository. It represents that whole services in the system should have shared information and data model and should not have the different data repository to deal with the same information. It can be obtained by building consolidated database, and all the services in the system should communicate each other by using the data from it.

4.3 Service Registry Management

In order to manage the services derived from the service map efficiently, Service Registry Management functionalities are required. It provides the functions such as

registering, updating and deleting the services to the service registry. It also provide authority management mechanism that keeps unauthorized user or system from finding and executing the important services. And, it need to keep an eye on whether the services is executed properly or not and log the result when the error occur. It also need to track and testing the execution time of each services and log the execution time of them. These logged results help system or operator making plans to use the services. And It also need to log user list and history for each services in order to prevent the services from being executed by too many systems and users at once.

4.4 Service Orchestration

If the functionalities to manage services are prepared, the business process can be developed by orchestration of services securely and efficiently. Figure 5 shows the SOA-based Next Generation OSS architecture which contains all the described features previously.

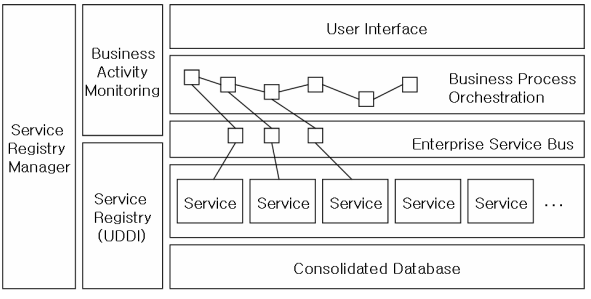


Fig. 5. SOA-based Next Generation OSS Architecture

5 Applying SOA-Based OSS to FMC Service

Wireless Broadband Internet (WiBro) service is a wireless mobile internet service, to be commercialized by KT in June 2005, for the first time in the world. WiBro is a leading example of FMC (Fixed Mobile Convergence) services which allows wireless access to broadband Internet. Figure 6 shows network topology of the service.

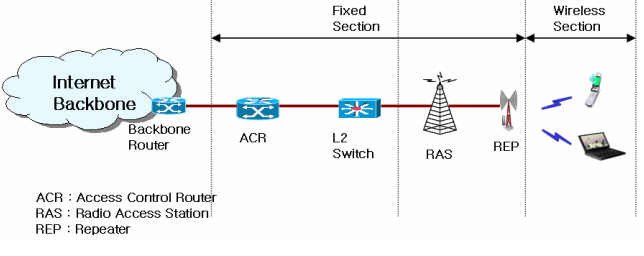


Fig. 6. WiBro Service Network Topology

WiBro requires the management of both fixed and wireless sections as shown in the network topology. When a resource trouble occurs, operators shall be able to locate in the assurance management view in which section the trouble occurred. Table 2 shows a sample of decomposed services of WiBro derived by using the aforementioned service map. Here, the service of '1.1.3.PD.DI Testing Resource Trouble' may be decomposed into two lower level services: device test functions for the fixed section and the wireless section, and then further be decomposed into test functions by equipment type in each section and services can be orchestrated and developed with the architecture which is shown in figure.5.

Table 2. The part of service map for WiBro Assurance Management

Level 1	Level 2	Level 3	Level 4	…
1.PI.DD Assuarance	1.1.PD.DI Resource Trouble Mgmt.	1.1.1.PD.DI Analyze Resource Trouble	…	…
		1.1.2.PD.DI Create Resource Trouble Report	…	…
		1.1.3.PD.DI Testing Resource Trouble	1.1.3.1.PI.DI Testing Fixed Section Resource	…
			1.1.3.2.PI.DI Testing Wireless Fixed Section Resource	…
…	…	…	…	…

During this process, if a testing function for the fixed section is already registered as a web service at the Service Registry, the test function module can be used during assurance management of WiBro. That is, a simple addition of a device testing function for the wireless section will be sufficient, so cost as well as development time will be reduced.

Also, by publishing the testing function as a web service, this function can be shared by other systems that also need such function. For example, as demonstrated in figure 7, it is assumed that an existing network management system in charge of Metro-Ethernet makes a device testing function for L2 switch and publishes it to the Service Registry. If WiBro service is implemented afterwards, in developing WiBro NMS, the desired Metro Switch testing function can be searched in the Service Registry for use. Other systems can always make use of this service from the Service Registry.

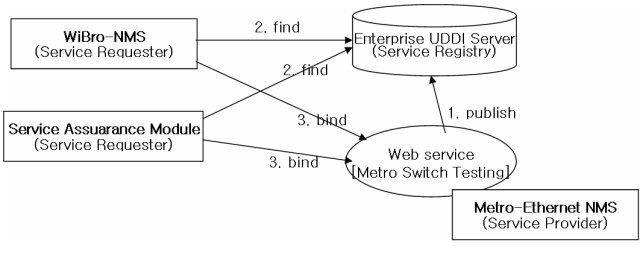


Fig. 7. An Example of Web Service [Metro Switch Testing]

6 Conclusion

This paper proposed SOA-based Next Generation OSS architecture which satisfies NGOSS key principles from TM Forum by applying the basic concepts of SOA. We suggested the service map which is the tool to derive services from operations and we described BPM which orchestrate services and develop business process, consolidated database which provides common communication language between each services, service registry manager which enables services can be used properly. And, this paper also presents adopting new convergence service will be easier than legacy system by reusing existing service with service repository with example of WiBro.

References

1. Roy W. Schulte Yefim V. Natis, "Service Oriented Architecture" Gartner Group, SSA Research Note SPA-401-068,1996
2. Randy Heffner, "The Big Strategic Impact Of Organic Business And Service-Oriented Architecture, Jun 2004
3. TMF, "Enhanced Telecom Operation Map (eTOM) – The Business Process Framework", GB921, V6.1 Nov 2005
4. TMF, "Shared Information and Data Model (SID), GB922, V6.1 Nov 2005
5. TMF, "The NGOSS Technology Neutral Architecture", TMF053 (and Addenda), Aug 2004
6. W3C Web Services WG, "Web Services Architecture", http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, Feb 2004
7. www.uddi.org, "UDDI Technical White Paper", Oct 2004
8. David Booth,Canyang Kevin Liu, "Web Services Description Language(WSDL) Version 2.0 Part 0: Primer", http://www.w3.org/TR/2006/CR-wsdl20-primer-20060327, Mar 2006
9. Nilo Mitra, "SOAP Version 1.2 Part 0:Primer", http://www.w3.org/TR/2003/REC-soap12-part0-20030624/, Jun 2003
10. Howard Smith, Peter Finger, "Business Process Management the third wave", Meghan-Kiffer press, 2003

Performance Analysis of a Centralized Resource Allocation Mechanism for Time-Slotted OBS Networks

Tai-Won Um[1], Jun Kyun Choi[2], Seong Gon Choi[3], and Won Ryu[1]

[1] Electronics and Telecommunications Research Institute,
161, Gajeong-dong, Yuseong-gu, Deajeon, 305-350, Korea
{twum, wlyu}@etri.re.kr
[2] Information and Communications University,
P.O. Box 77, Yusong, Daejon, 305-348, Korea
jkchoi@icu.ac.kr
[3] Chungbuk National University
12 Gaeshin-dong, Heungduk-gu, Chungbuk, Korea.
sgchoi@chungbuk.ac.kr

Abstract. Time-Slotted Optical Burst Switching (TS-OBS) is one of the most promising next-generation transport network technologies. This paper proposes a time-slot assignment procedure using centralized control for TS-OBS networks. Our scheme attempts to improve burst contention resolution and optical channel utilization. Analysis and simulation results show that the channel utilization of the TS-OBS network is improved markedly at the expense of ingress buffering delay.

Keywords: time-slotted optical burst switching, centralized resource allocation.

1 Introduction

Optical burst switching (OBS) [1], [2] is a novel optical switching paradigm, which is capable of enhancing optical channel utilization by multiplexing collected packets, referred to as an optical burst, onto a wavelength. There have been a number of OBS reservation protocols, such as just-enough-time (JET) [1], just-in-time (JIT) [3], Horizon, and wavelength-routed OBS (WR-OBS) [4], [5].

In JET, an ingress node transmits a control packet through a dedicated control channel in order to configure each OBS switch for the burst duration along the paths before sending out its corresponding optical burst. The control packet and optical burst are separated by a certain delta time called offset, which is required because the control packet is electronically processed at intermediate switches while the optical burst is not. The OBS inevitably suffers from an optical burst loss problem due to a contention for obtaining optical resources between control packets within an intermediate node if the number of burst simultaneously arrived at an output port of an OBS switch is more than the number of available channel.

In WR-OBS, while an ingress node aggregates packets into a burst, it sends a wavelength request to a control node. When an acknowledgement is received, the burst is assigned to an available wavelength-routed path. Compared to conventional JET, this scheme is able to support explicit QoS provisioning and deterministic delay

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 403 – 411, 2006.
© Springer-Verlag Berlin Heidelberg 2006

for the optical burst, but it wastes wavelength resource during signaling because intermediate nodes begin reserving the whole wavelength for that request when the acknowledgement arrives there. Moreover, as the burst size decreases and signaling delay for establishing a wavelength-routed path increases, channel utilization decreases.

On the other hand, Time Slotted Optical Burst Switching (TS-OBS) [6],[7] is a form of OBS where time is essentially quantized into discrete units, referred to as slots. Obviously, in TS-OBS based on synchronous transmission the chance of contention is smaller than OBS based on asynchronous transmission because the behavior of the bursts is more predictable and regulated [8].

This paper describes a slot assignment procedure using centralized control for TS-OBS networks. In this scheme the controller is responsible for finding a path, configuring time-slotted optical cross connects (ts-OXCs) [9],[10] and assigning a time-slot to transmit an optical burst. Our scheme intends to improve burst contention resolution and optical channel utilization. The rest of the paper is organized as follows. In Section 2, we describe the TS-OBS network architecture with centralized control, and Section 3 focuses on the centralized resource allocation mechanism. The results obtained from analysis and simulations will be discussed in Section 4. Finally, we draw our conclusions in Section 5.

2 Network Architecture

Network control methods can be classified into three categories: centralized, distributed and hybrid control. Each control method have pros and cons, but typical telecommunication networks and automatic switched optical networks (ASON) [11] in ITU-T follow the centralized control architecture, in that the control plane is separated from the data plane.

In this paper, we consider the centralized control architecture for TS-OBS networks in order to execute the path calculation and slot assignment. As shown in Fig 1, the control plane consisted of controllers is separated from the data plane consisted of ts-OXC, and there is an interface channel between a controller and a ts-OXC, which can communicate each other by defining signaling messages.

In this TS-OBS network, ts-OXC is made up of all-optical switching components, operates on a synchronized timing basis without O/E/O conversion. The wavelengths on the optical links are split into time-slotted wavelengths of a fixed-size divided by a time interval. One or more time-slot(s) can be assigned for the request from the ingress ts-OXC to transmit an optical burst. All the link propagation delays are an integer number of time slots. On the path from the ingress ts-OXC to the egress ts-OXC, each ts-OXC has a role in switching incoming time-slots to outgoing time-slots in order to transport the optical burst. In order to do this, every ts-OXC needs to maintain switching tables containing information entries that are received from the controller, to forward slotted wavelengths.

The network topology information will be integrated to the controller by routing protocols. By applying the time-slot assignment mechanism into the routing table, the controller may find a number of paths. One path could be selected according to constraints given for the request.

If the number of ts-OXCs that are managed by a centralized controller increases, control and configuration of paths will be a burden to the controller. Under a controller, the number of ts-OXCs will be limited according to its processing power. The controllers are connected to each other and form a control network. If an ingress ts-OXC and an egress ts-OXC exist under different controllers, the corresponding controllers should share its topology. The path information will be sent from the controller managing the ingress ts-OXC to the controller managing the egress ts-OXC. After the path calculation, each controller on the path should send the forwarding table to the corresponding ts-OXCs to configure their switch.

In order to exchange of control and topology information between a controller and ts-OXCs and between controllers, a new protocol may be proposed or the generalized multiprotocol label switching (GMPLS) protocol could be extended. As control networks increase, it could be built hierarchically. The configuration method of the control network is out of the scope of this paper.

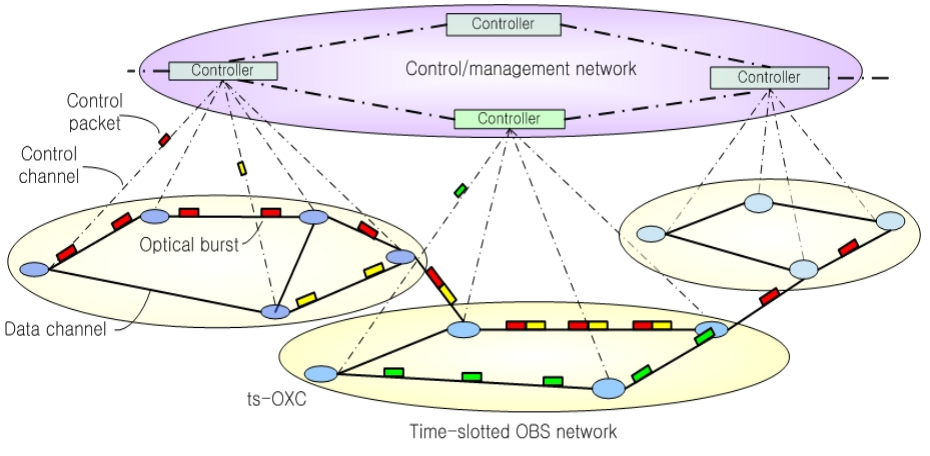


Fig. 1. Network architecture using centralized control

3 Centralized Resource Allocation Mechanism

A request for a path calculation and slot assignment from an ingress ts-OXC is delivered to its corresponding controller. Upon receiving the request, the controller takes the responsibility of finding an optimal path to transport the optical burst on a TS-OBS network, delivering the configuration information to intermediate ts-OXCs and informing the assigned time-slot to the ingress ts-OXC in order to transmit an optical burst.

Fig. 2 shows a centalized resource allocation and burst transmission procedure on demand of the control packet sent from an ingress ts-OXC. When a data packet arrives at an ingress ts-OXC of the TS-OBS network, the ingress ts-OXC will look up a packet's egress ts-OXC and push it to the corresponding queue to the destination. While the packets are aggregating in the queue, if the queue size reaches a given threshold or timeout signal for delay-sensitive data, it has to send a control packet to the controller to request an assigned time-slot for the optical burst. The control packet

has to be processed by the routing, wavelength and timeslot assignment (RWTA) algorithm in the controller [12]. Similarly as described in [5], the controller may estimate the traffic arrival time from the packet accumulated when the control packet is sent and, hence, establishes the fixed burst size by the time the acknowledgement arrives back at the sending ingress ts-OXC.

When the controller receives the control packet, it decides whether it can accept this request or not. If, in the requested time-slot, all slotted wavelengths have been reserved for other requests, then the control packet will be rejected. If there is an available slot, it will reserve the time slot and reply to the ingress ts-OXC with an acknowledgement packet. When the ingress ts-OXC receives the acknowledgement packet, it can transmit the optical burst to the core ts-OXC via the assigned time-slot. If the control packet is rejected, the ingress ts-OXC may resend the control packet to reserve the time slot after random time-lag.

In order to reduce processing delay at a controller, it is recommended that the controller find a path before receiving a slot-assignment request. Key factors to minimize the ingress delay are how fast a controller calculates a path, assigns a time-slot without slot-contention and informs it to the controller.

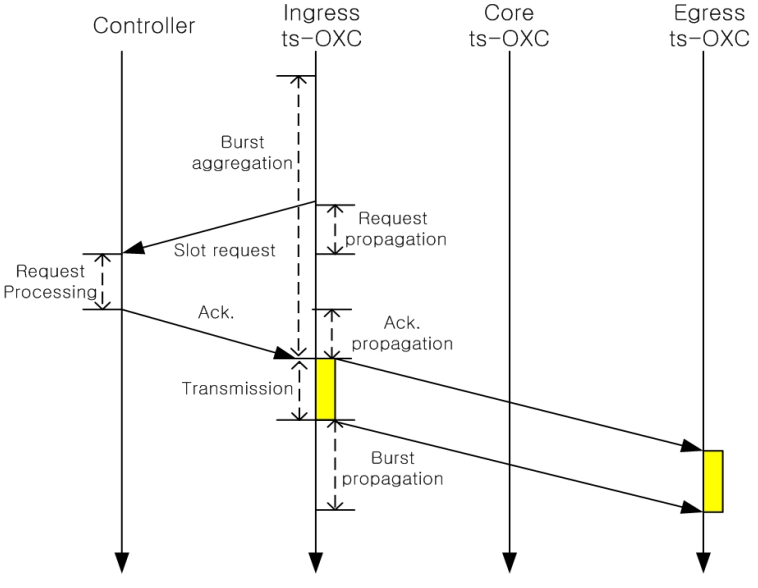


Fig. 2. Centalized resource allocation and burst transmission procedure

4 Performance Analysis

4.1 Analysis of Channel Utilization

To determine the channel utilization for our proposed scheme, we extend the analytical model introduced in [5] for WR-OBS networks. For clarity and simplicity, the analysis in this section is based only on mean values for all parameters.

The burst aggregating cycle can be explained as follows. Let the ingress delay $t_{ingress}$ be the burst aggregation time. We define $t_{prop,control}$ to be the propagation delay for the control packet. Processing the time-slot request requires time t_{proc}, followed by an acknowledgment packet to be returned to the requesting ingress node, with an additional delay $t_{prop,ack}$. After that, a wavelength channel is allocated during slot time t_{slot}. Actually, there is a guard-time (t_{guard}) between the time-slotted wavelengths because each intermediate optical switch requires switching time and a slot can be interfered with by an adjacent slot by dispersion and non-linear effects on the physical layer. So, t_{WHT} consists of t_{slot} and t_{guard}. In slot time t_{slot} , a burst in the buffer of an ingress ts-OXC is sent. Let the bit rate b_{in} be the aggregated bit rate for the traffic from all n sources directed to a particular destination and requiring the same QoS. Bursts are transmitted from the queue at core bit rate b_{core}, where $b_{core} > b_{in}$. The number of wavelengths is $A = b_{core} / b_{in}$. This time is related with $t_{trans} = L_{burst}/b_{core}$, which is the time to complete the burst transmission. We assume that t_{slot} and t_{trans} are the same to simplify this analysis. The propagation delay, t_{prop}, is a delay to propagate bursts for receiving an egress ts-OXC.

The maximum deterministic latency or upper bound on the maximum transmission time that packets experience between entering the core network at the ingress ts-OXC and egress ts-OXC is

$$Latency_{\max} = t_{ingress} + t_{WHT} + t_{prop}.$$

The arrival of the acknowledgment packet from the controller sets the start of the subsequent burst aggregation and the cycle repeats. From the analysis of the timing involved in burst aggregation and transmission, it is clear that network efficiency depends on the processing speed of the network controller.

It is assumed that burst sizes increase linearly, equivalent to the case of CBR traffic arriving to the buffer. Thus, for a constant load and CBR traffic, the burst size L_{burst} is proportional to the ingress delay and the input bit-rate b_{in}, so that $L_{burst} = b_{in} \cdot t_{ingress}$.

A parameter following from [5] is the bandwidth per wavelength, which indicates the effective bandwidth of a lightpath used for transmission of data between edge ts-OXCs.

$$B_{per\,slot} = \frac{L_{burst}}{t_{WHT}} = \frac{b_{core} * t_{slot}}{t_{guard} + t_{slot}}$$

The influence of the guard time is shown in Fig. 3(a) for b_{core} = 3 Gbps, and t_{guard} = 0.15 ms, 1.5 ms, 15 ms. The increase in bandwidth for the identical values t_{slot} is reduced for higher t_{guard}; for t_{guard} = 15 ms, values remain below 2.6 Gbps for a 3 Gbps physical bit rate. Fig. 3(b) shows the effect of bandwidth saturation for t_{guard} = 1.5 ms and core bit rates vary from 1 to 5 Gbps. The significance of the results is that $B_{per\ slot}$ remains significantly smaller than the optical line rate for $t_{slot} \leq 40/A$ ms, because the packet loss probability is too high to send the burst when $t_{ingress}$ is bigger than 40ms.

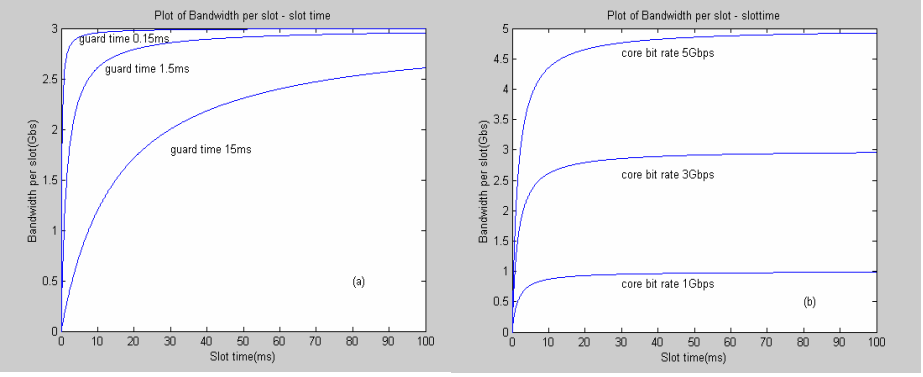


Fig. 3. Bandwidth per slot B_{per} slot for (a) b_{core} = 3 Gbps, and t_{guard} = 0.15ms, 1.5ms, 15ms and for (b) t_{guard} = 1.5ms, b_{core} = 1 Gbps, 3 Gbps, 5 Gbps

4.2 Simulation Results

To analyze the proposed TS-OBS scheme, we have developed an OBS simulator by extending the NS2 simulator. This OBS simulator consists of an ingress node, a core node, an egress node and a controller. The ingress node aggregates packets into a burst and generates a control packet to the corresponding controller. Upon receiving the control packet, the controller compares the requested time-slot with previous burst reservations based on burst arrival record tables. If there is an available slot, it will reserve the time slot and reply to the ingress ts-OXC with an acknowledgement packet. When the ingress ts-OXC receives the acknowledgement packet, it can transmit the optical burst to the core ts-OXC via the assigned time-slot. If the control packet is rejected, the ingress ts-OXC resends the control packet to reserve a time slot.

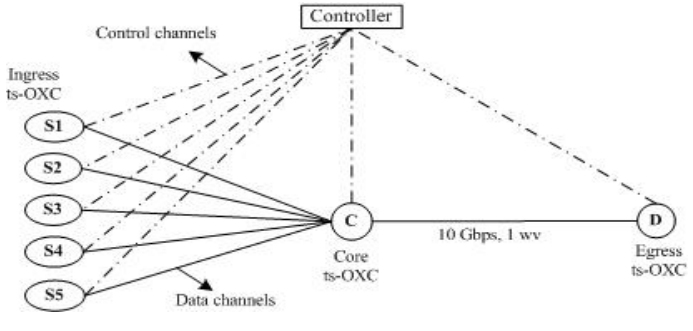


Fig. 4. Simulation Topology

The network topology and parameters for the simulation are given in Fig. 4. We consider 5 ingress ts-OXCs, 1 core ts-OXC and 1 egress ts-OXC, which support full wavelength conversion. We assume that the average arrival rate is the same for all

ingress nodes. Packets arrive at each ingress node according to a Poisson process with a 2Gbps input bit rate. Packets are aggregated into a burst of 12.5kbyte in size at the ingress node, and sent to the egress ts-OXC. The performance metrics are ingress queueing delay, buffer size and link utilization as a function of offered input traffic load.

Fig. 5 shows the link utilization of our TS-OBS scheme using a centralized slot assignment as a function of the offered traffic load per ingress node. Comparing the link utilization of conventional OBS with the proposed scheme shows that our scheme improves the utilization markedly. In conventional OBS, if optical bursts collide with one another, they will be dropped. However, in our scheme, by sending the control packet to the centralized controller, the ingress ts-OXC can prove the available time-slots, and if there are no available time-slots, the ingress node will try to reserve the next slot again until it succeeds, instead of dropping the optical burst.

Fig. 6 shows the ingress buffering delay versus the offered load when the propagation delay from the ingress node to the controller is 2ms, 4ms or 6ms. In the proposed scheme, if there is not an available time-slot, the buffered data will wait until it reserves a time-slot. The signaling time to reserve a time-slot from the ingress node to the controller depends on the distance between them. Therefore if the propagation delay to the controller increases, the assembled packets experience more ingress delay, and the ingress node needs more buffers (as shown in Fig. 7).

In conventional OBS, when a burst is blocked, the only way to recover the burst is through TCP retransmission. In order to achieve a connection oriented session, TCP makes use of retransmission on timeouts and positive acknowledgments upon receipt of information. However, TCP does not provide fast recovery due to its host-to-host behavior and time-out mechanism. On the other hand, UDP is connectionless, which means that it can not provide error control and flow control. Therefore, if we consider the TCP/UDP layer's retransmission of lost packets in conventional OBS, buffering at the OBS layer in our scheme may support better performance for upper layers. In this paper, we do not analyze the performance of the TCP layer; that remains for future studies.

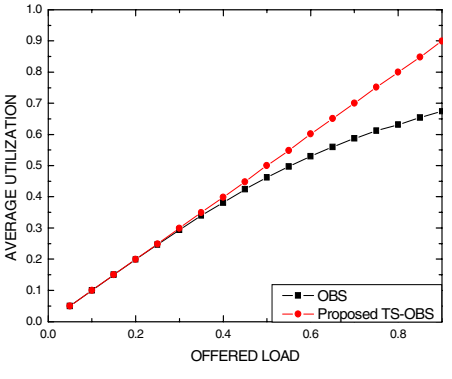


Fig. 5. Offered load vs. Link utilization

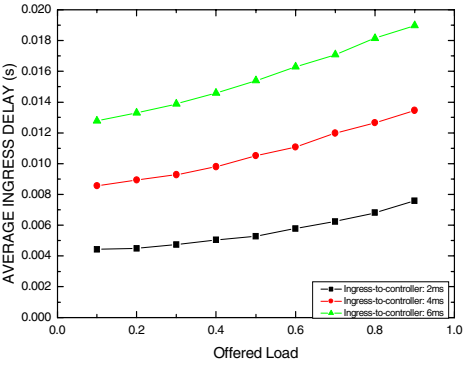


Fig. 6. Offered load vs. Ingress buffering delay

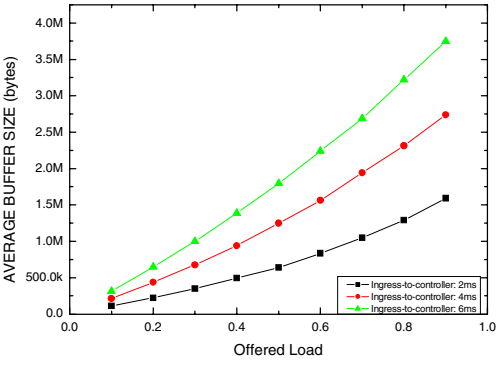


Fig. 7. Offered load vs. Average buffer size

5 Conclusion

In this paper, we proposed a centralized control architecture and time-slot assignment procedure for TS-OBS networks. In this architecture, a request of path calculation and slot assignment from an ingress ts-OXC is delivered to a controller. Upon receiving the request, the controller takes the responsibility of finding an optimal path, delivering the configuration information to ts-OXCs and informing the assigned time-slot to the ingress ts-OXC. Analysis and simulation results show that the channel utilization of the TS-OBS network is improved markedly at the expense of ingress buffering delay.

Acknowledgments. This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ministry of Science and Technology (MOST); and the Institute of Information Technology Assessment (IITA) through the Ministry of Information and Communication (MIC), Korea.

References

1. M. Yoo, C. Qiao and S. Dixit, “QoS Performance of Optical Burst Switching in IP-Over-WDM Networks,” IEEE JSAC, vol. 18, no. 10, pp. 2062-2071, Oct. 2000.
2. C. Qiao, “Labeled Optical Burst Switching for IP-over-WDM Integration," IEEE Communication magazine, vol. 38, no. 9, pp. 104-114, Sep. 2002.
3. J. Y. Wei and R. I. McFarland, “Just-In-Time Signaling for WDM Optical Burst Switching Networks,” IEEE JLT, vol. 18, no. 12, pp. 2109–2031, Dec. 2000.
4. Michael Duser, Polina Bayvel, “Performance of a Dynamically Wavelength-Routed Optical Burst Switched Network”, IEEE Photonics technologies letter, vol. 14, no. 2, pp. 239-241. Feb. 2002.
5. Michael Duser, Polina Bayvel, “Analysis of a Dynamically Wavelength-Routed Optical Burst Switched Network Architecture”, Journal of lightwave technologies, vol. 20, no 4, pp. 573-585. Apr. 2002.
6. Jeyashankher Ramamirtham and Jonathan Turner, “Time Sliced Optical Burst Switching,” IEEE INFOCOM’03, pp. 2030-2038, Apr. 2003.

7. Geoffrey M. Garner, "Advanced Plesiochronous TSOBS," prepared for SAIT, Dec. 2004.
8. S. Yao, S. J. B. Yoo and B. Mukherjee, "A comparison study between slotted and unslotted all optical packet-switched network with priority-based routing," OFC'01, Mar. 2001.
9. I.P. Kaminow, et al., "A Wideband All-Optical WDM Network", IEEE Journal on Selected Areas in Communications. Jun. 1996.
10. Nen-Fu Huang, et al., "A Novel All-Optical Transport Network with Time-Shared Wavelength Channels", IEEE Journal on Selected Areas in Communications, Oct. 2000.
11. "Architecture for the automatically switched optical network (ASON)", ITU-T Recommendation, G.8080/Y.1304, Nov. 2001.
12. Bo Wen, Krishna M. Sivalingam, "Routing, Wavelength and Time-Slot Assignment in Time Division Multiplexed Wavelength-Routed Optical WDM Networks," IEEE INFORCOM 2002.
13. Panos Trimintzios, et al., "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks," IEEE Comm. Mag., May 2001.
14. Jun Kyun Choi, et al., "Fast Protection Mechanism Using Centralized Control with SRLG", IETF draft, draft-choi-centralized-protection-00.txt, Jul. 2004.

Efficient Performance Management of Subcarrier-Allocation Systems in Orthogonal Frequency-Division Multiple Access Networks

Jui-Chi Chen[1] and Wen-Shyen E. Chen[2]

[1] Department of Computer Science and Information Engineering, Asia University
500, Lioufeng Rd., Wufeng, 41354 Taichung, Taiwan
rjchen@ieee.org
[2] Department of Computer Science, National Chung-Hsing University
250, KuoKuang Rd., 40227 Taichung, Taiwan
echen@cs.nchu.edu.tw

Abstract. In next-generation wireless networks, OFDM is a potential key technology. Subcarriers in the OFDMA networks are scarce and valuable, operators should manage them efficiently by radio-resource management function. Several OFDMA subcarrier-allocation schemes and performance-evaluation methods have been widely studied, but very few studies focused on managing call blocking probability (CBP) and bandwidth utilization (BU). CBP and BU are two important performance measures, which respectively represent the quality of service for subscribers and the profit of operators. This paper proposes an evaluation model to accurately predict the subcarrier-allocation performance, especially in consideration of the CBP and BU. Theoretical and simulation results indicate that the model works. In addition, the CBP and BU are adopted to solve a subcarrier utilization-optimization problem, which can be applied in network planning and re-planning processes. The optimization problem uses given traffic load and specified maximum CBP constraint to maximize the BU by discovering the optimal number of subcarriers in a base station. Finally, operators may utilize the results to manage their subcarriers flexibly, and to improve their profitability.

Keywords: Performance management; OFDMA; Subcarrier allocation; Utilization optimization.

1 Introduction

Next-generation wireless networks even provide high capacity and high data-rate services, enabling high quality images and video to be transmitted and received. Orthogonal frequency-division multiplexing (OFDM) is a potential key technology for the next-generation broadband wireless networks [1,2]. It has several advantages, such as flexibility of allocating subcarriers to users, high spectral efficiency, low receiver complexity and simple implementation by inverse fast Fourier transform (IFFT) and FFT [3,4].

Subcarriers in the orthogonal frequency-division multiple access (OFDMA) networks are scarce and valuable, so operators should employ them efficiently by the

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 412 – 421, 2006.
© Springer-Verlag Berlin Heidelberg 2006

radio-resource management function, which controls the subcarrier allocation, maintenance and termination. Recently, several studies have proposed efficient OFDMA subcarrier-allocation schemes [5–10] and performance-evaluation methods [11–14] widely. However, very few studies focused on managing call blocking probability (CBP) and bandwidth utilization (BU). CBP and BU are two important performance measures in the OFDMA subcarrier-allocation system. CBP denotes the probability of blocking incoming call requests, and represents the quality of service for subscribers. Additionally, an operator has to utilize the scarce and valuable subcarriers for raising BU, which is the rate of total bandwidth used during a period of time, to increase profit. This paper proposes a batch-arrival queueing model for evaluating the performance of the subcarrier-allocation system, especially in consideration of the CBP and BU. The number of states in the model rises only linearly with the increasing number of subcarriers, which can therefore be applied to solve the equilibrium equations. System capacity, the number of subcarriers in a base station, is also restricted by the allocated frequency bandwidth, so that a new call or service request may be rejected if the cell has insufficient capacity when the request arrives. The rejected call is also called a blocked call. Then we use the CBP and BU measures to solve a subcarrier utilization-optimization problem. The optimization problem finds the optimal number of subcarriers in a base station or a cell to maximize the BU with a given traffic load and a specified maximum CBP constraint.

The remainder of this study is organized as follows. Section 2 describes the OFDMA technique. Section 3 presents the proposed model, while Section 4 shows the simulation results and verifies the theoretical analysis. Section 5 applies the performance measures to solve the subcarrier utilization-optimization problem. Conclusion is finally drawn in Section 6.

2 Orthogonal Frequency Division Multiple Access

Traditional frequency-division multiplexing (FDM) simultaneously sends multiple signals over one transmission path. Each signal travels within its own frequency carrier. OFDM distributes the data over a large number of subcarriers that are spaced apart at precise frequencies. This spacing provides the orthogonality, which prevents the demodulators from seeing other frequencies [1,2]. OFDM squeezes multiple subcarriers tightly together to obtain a high spectral efficiency, but still keeps the modulated signals orthogonal to ensure that the subcarriers do not interfere with each other. Fig. 1 indicates that OFDM permits signals to overlap, thus significantly reducing its required bandwidth. OFDMA enables some subcarriers to be assigned to different users; for instance, subcarriers 1, 3, 5 and 7 can be assigned to user A, and subcarriers 2, 4, 6 and 8 to user B. OFDM can also be integrated with multiple-input multiple-output (MIMO) technique to increase the diversity gain and/or achieve capacity gain [4,5]. MIMO applies multiple antennae to transmit data in small pieces to the receiver, which can reconstruct the data flows. This process is called spatial multiplexing.

However, subcarriers support various wideband services with different data rates. Both the uplink and the downlink in OFDM can use several subcarriers in parallel with various adaptive modulation and coding (AMC) parameters to match a

user-request data rate. The available bandwidth in a base station is limited and shared if the values of AMC parameters are decided and used. Every user shares the FFT uplink space, and the base station assigns subcarriers to users [3]. Without loss of generality, assume that a cell (a base station may contain one or a few cells) owns a subcarrier-allocation system.

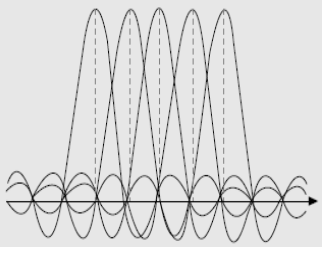


Fig. 1. Example for orthogonality among all subcarriers [15]

3 Evaluation Model for the OFDMA Subcarrier-Allocation System

From the point of view of analytical purpose, we may obtain the statistical average values of AMC parameters for each subcarrier in advance. Restated, every subcarrier has the same average data rate. The number of subcarriers c generally denotes the system capacity in a cell. That is, the cell has in total cR_b rate resources, where R_b represents the average data rate per subcarrier. A user (call) can request multiple subcarriers to fulfill its transmission requirement. For a system, that a user requesting kR_b (k subcarriers) arrives can be seen as that k users, each of them requests R_b, arrive simultaneously. Hence this case is considered as a batch (group; bulk) arrival with size k. Assume that the users arrive in groups following a Poisson process with mean group-arrival rate λ. The system probability sequence $\{x_k\}$ governs the size of the arriving group. Restated, an arriving group has size k and probability x_k. Let λ_k denote the batch arrival rate with the group size of Poisson user stream k, where $\lambda_k = x_k \lambda$, $\sum_{k=1}^{n} x_k = 1$, $1 \le k \le n \le c$, and $k, n \in N$. The variable n represents the highest data rate nR_b that the system can provide. Then the average group size equals $\hat{g} = \sum_{k=1}^{n} k x_k$. Furthermore, assume that the service time (call holding time) for all users is independently exponentially distributed with parameter $1/\mu$. Thus the system can be viewed as a multi-channel queue, with c parallel servers with rate μ.

Fig. 2 depicts the flow-in flow-out diagram for state m on the subcarrier-allocation system, where $n \le m \le c - n$. The incoming flows include $(m + 1)\mu$ from state $m + 1$ and all possible batch arrivals λ_k with size k, where $1 \le k \le n$. From the flow-out point of view, state m has rate $m\mu$ to flow to state $m - 1$; λ_1 to state $m + 1$; λ_2 to state $m + 2$; λ_3 to state $m + 3$; λ_k to state $m + k$, and λ_n to state $m + n$.

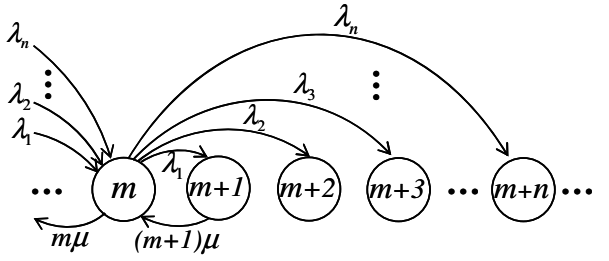


Fig. 2. Demonstration of the flow-in flow-out diagram for state m on the OFDMA subcarrier-allocation system

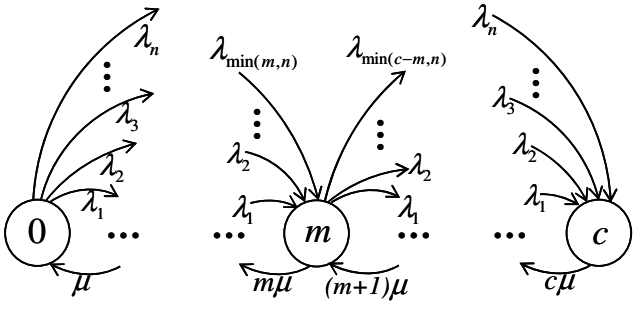


Fig. 3. The state-transition-rate diagram of $M^X/M/c/c$ for the OFDMA subcarrier-allocation system

Moreover, the system has no buffer (the queue length equals zero), making it a loss system. Thus, the subcarrier-allocation system can be modeled on the batch-arrival model $M^X/M/c/c$, whose state-transition-rate diagram is shown in Fig. 3, where $1 \le m \le c - 1$. If $c \ge n$, then state 0 has the flow-out rate $\lambda = \sum_{k=1}^{n} \lambda_k$. When $m < n$, the maximum batch-arrival size from some previous state is given by m. Thus, the maximum rate of incoming arrivals into state m should be $\lambda_{\min(m,n)}$. If $m \le c - n$, then all outgoing arrivals from state m have the same case, as illustrated in Fig. 2. However, the arrivals only have partial success in being served with smaller size than $c - m + 1$ when $c - n < m \le c - 1$. Hence, the outgoing arrival rates from state m are given by $\lambda_{\min(c-m,n)}$.

By queueing theory, the equilibrium (steady-state) equations written below are run to obtain the steady-state probabilities p_m of the model.

$$\lambda p_0 = \mu p_1, \text{ where } 1 \le n \le c \text{ and } \lambda = \sum_{k=1}^{n} \lambda_k . \tag{1}$$

$$\left(m\mu + \sum_{k=1}^{\min(c-m,n)} \lambda_k \right) p_m = \sum_{k=1}^{\min(m,n)} \lambda_k p_{m-k} + (m+1)\mu p_{m+1}, \text{ where } 1 \le m \le c-1 . \tag{2}$$

$$c\mu p_c = \sum_{k=1}^{n} \lambda_k p_{c-k}, \tag{3}$$

which can be used for verification. Reforming (1) and (2) yields

$$p_1 = p_0 \, \lambda/\mu \text{, and} \tag{4}$$

$$p_{m+1} = \left[\left(m\mu + \sum_{k=1}^{\min(c-m,n)} \lambda_k \right) p_m - \sum_{k=1}^{\min(m,n)} \lambda_k p_{m-k} \right] \Big/ (m+1)\mu \text{, where } 1 \le m \le c-1. \tag{5}$$

Recursive programs cannot always solve the equations, owing to overabundant recursive levels for large *c*. Therefore, an iterative procedure is adopted to solve the equilibrium equations as follows.

$$\text{Let } p_0^* = 1 \text{; then } p_1^* = p_0^* \, (\lambda/\mu) = \lambda/\mu. \tag{6}$$

$$p_{m+1}^* = \left[\left(m\mu + \sum_{k=1}^{\min(c-m,n)} \lambda_k \right) p_m^* - \sum_{k=1}^{\min(m,n)} \lambda_k p_{m-k}^* \right] \Big/ (m+1)\mu \text{, where } 1 \le m \le c-1. \tag{7}$$

According to the normalizing condition $\sum_{i=0}^{c} p_i = 1$, the equilibrium probabilities of all states are written as follows:

$$p_m = p_m^* \Big/ \sum_{i=0}^{c} p_i^* \text{, where } 0 \le m \le c. \tag{8}$$

Once the equilibrium state probabilities are known, the CBP and BU can be derived forwardly. If a new call finds that the available capacity in the corresponding cell cannot meet its rate requirement, then it is fully blocked. CBP is expressed as the number of blocked calls divided by the number of total calls during a period of time. Thus the CBP of the batch-arrival model can be written as

$$\Omega = \sum_{i=0}^{n-1} \left(p_{c-i} \sum_{k=i+1}^{n} \lambda_k \Big/ \lambda \right), \text{ where } 1 \le n \le c, \text{ and } \lambda = \sum_{k=1}^{n} \lambda_k. \tag{9}$$

Furthermore, the average number of customers (average system length; ASL) in the system is given by

$$L_s = \sum_{i=0}^{c} i p_i. \tag{10}$$

L_s equals the mean number of busy servers in the system, because the queue size is zero. Observing the system for a long period of time *T*, we have the average BU shown below.

$$\Psi_1 = \lim_{T\to\infty} \frac{\sum_{k=1}^{n}\left[\lambda_k T(1-\sum_{i=c-k+1}^{c} p_i)\cdot k\cdot\frac{1}{\mu}\right]}{T\cdot c} \quad \text{L'Hospital} \quad = \quad \frac{\sum_{k=1}^{n}\left[k\lambda_k(1-\sum_{i=c-k+1}^{c} p_i)\right]}{c\mu}, \tag{11}$$

where $\sum_{k=1}^{n}\left[k\lambda_k(1-\sum_{i=c-k+1}^{c} p_i)\right] = \hat{g}\lambda_{eff}$; $1 \le n \le c$, and $1/\mu$ is the mean call holding time. Period T has $\lambda_k T$ calls with kR_b incoming, where $n \ge k \ge 1$, and thus a total of $\lambda_k T(1-\sum_{i=c-k+1}^{c} p_i)$ rate-k calls are served. The rate-k bandwidth usage is given by $\lambda_k T(1-\sum_{i=c-k+1}^{c} p_i)\cdot k$. Therefore, $\sum_{k=1}^{n}\left[\lambda_k T(1-\sum_{i=c-k+1}^{c} p_i)\cdot k\cdot\frac{1}{\mu}\right]$ represents the total usage time of all possible successful users in the period T, where $1/\mu$ is the mean call holding time. The numerical analysis demonstrates that

$$\Psi_2 = L_s / c = \Psi_1 \ . \tag{12}$$

4 Theoretical and Simulation Results

Various performance measures for the proposed model were measured for two cases in which the arriving group size (call-request data rate) has a discrete uniform distribution and a geometric distribution. Any countable distributions, e.g., constant, discrete uniform, and geometric distributions, can be applied to map the behavior of the arriving group size. The simulation defines the CBP as the number of blocked calls divided by the total number of calls, and the BU as the average consumed bandwidth divided by the total capacity. The system has offered load $\hat{g}\lambda/\mu$ and traffic intensity $\rho = \hat{g}\lambda/c\mu$. Assume that the downlink is based on a MIMO-OFDM air interface, with average coding rate 2/3, 16-QAM modulation, spatial multiplexing mode, and two transmit and up to four receive antennas, as shown in [4].

Given a carrier with a 20MHz bandwidth, the system has the peak throughput 82.4Mbps and divides the carrier to 128 subcarriers. Then each subcarrier supports the average data rate 659.2kbps. The CBP and BU were simulated and evaluated with various values of ρ ranging from 0.515625 to 3.09375, where c = 128; μ = 0.00625, and λ ranges from 0.025 to 0.15. The arriving group size was distributed with a discrete uniform distribution, where the maximum group size was 32 (n = 32), and the average arriving group size $\hat{g}$ = 16.5, that is, $\lambda_1 = \lambda_2 = \lambda_3 = \ldots = \lambda_{32} = \lambda/32$. Such small λ and μ values were adopted to obtain accurate simulation results. Surely, high values of λ and μ with the same ratio still result in the same outcome for theoretical analysis. Fig. 4 compares the theoretical and simulated CBP and BU results, where the horizontal axis denotes the traffic intensity given by $\rho = \hat{g}\lambda/c\mu$, and the vertical axis shows the CBP and BU. The theoretical CBP and BU values are shown as solid lines with squares and circles, while the simulation CBP and BU values are depicted

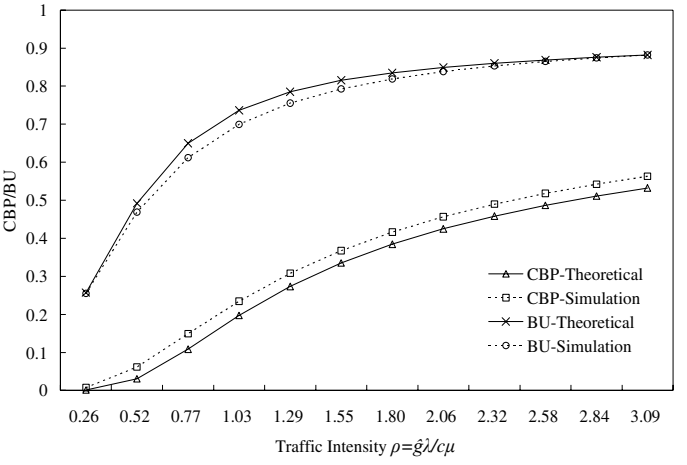


Fig. 4. Performance comparison between theoretical and simulation CBP and BU results, where $c = 128$; $n = 32$, and $\hat{g} = 16.5$

Table 1. Performance comparison between theoretical (*theo.*) and simulation (*simu.*) results by a geometric group-size distribution, where $c = 32$; $n = 16$, and $\hat{g} = 4.83682$

	CBP		BU			ASL	
	theo.	*simu.*	*theo.*	theo.	*simu.*	*theo.*	*simu.*
ρ	(Ω)		(Ψ_1)	(Ψ_2)		(L_s)	
0.45	0.0374	0.0516	0.4175	0.4175	0.3839	13.4	12.2
0.76	0.1118	0.1284	0.5902	0.5902	0.5495	18.9	17.5
1.06	0.1892	0.2050	0.6889	0.6889	0.6538	22.0	20.9
1.66	0.3121	0.3277	0.7872	0.7872	0.7674	25.2	24.5
1.96	0.3589	0.3740	0.8147	0.8147	0.7996	26.1	25.5
2.57	0.4328	0.4465	0.8509	0.8509	0.8424	27.2	26.9
2.87	0.4626	0.4754	0.8635	0.8635	0.8572	27.6	27.3
3.48	0.5120	0.5225	0.8826	0.8826	0.8791	28.2	28.0
3.63	0.5226	0.5324	0.8865	0.8865	0.8833	28.4	28.2

as dotted lines with cross signs and plus signs. As shown in Fig. 4, both the CBP and the BU rise with increasing ρ because of the limited capacity, making the simulation results close to the theoretical results.

In the second simulation, given a carrier with a 5MHz bandwidth, the system has the peak throughput 20.6Mbps and divides the carrier to 32 subcarriers. Then each subcarrier supports the average data rate 659.2kbps. Assume that the arriving group size has a geometric distribution, where $c = 32$; $n = 16$; $\hat{g} = 4.83682019$, and $\{x_k. \mid 1 \le k \le 16\} = \{0.18899, 0.15475, 0.12663, 0.10372, 0.08486, 0.06949, 0.05698, 0.04659, 0.03812, 0.03127, 0.02558, 0.02094, 0.01713, 0.01404, 0.01149, 0.00941\}$. Table 1 lists some of the simulation and theoretical results, where $\mu = 0.00625$ and λ ranges from 0.01875 to 0.15. The table also compares other theoretical measures with those

obtained in the simulation, including the theoretical bandwidth utilization Ψ_1 and Ψ_2, and the average number of busy servers L_s, and also presents the approximate result between the theoretical analysis and the simulation.

5 Subcarrier Utilization Optimization

In network planning or re-planning process, operators can efficiently find how many subcarriers in a cell result in the maximum utilization with a given CBP constraint. The results conducted from the proposed model can be applied to the utilization-optimization problem to maximize the profit under a specific CBP constraint by determining the optimal number of subcarriers, which should be recommended to the operators. Correspondingly, the operators allocate enough frequency bandwidth to support the optimal number of subcarriers for each cell. With given traffic load $\hat{g}\lambda/\mu$ and specified maximum CBP constraint $CBP_{threshold}$, the optimization problem can be expressed as follows.

Given λ, μ and $CBP_{threshold}$, determine the optimal number of subcarriers c so as to

$$\text{Maximize} \quad \frac{1}{c}\sum_{i=0}^{c} ip_i$$

$$\text{Subject to} \quad \lambda > 0, \mu > 0, 1 > CBP_{threshold} > 0$$

$$\sum_{i=0}^{n-1}\left(p_{c-i}\sum_{k=i+1}^{n}\lambda_k \Big/ \lambda \right) \le CBP_{threshold}, \text{ where } c, n \in N. \tag{13}$$

The monotonicity of CBP and BU in the subcarrier-allocation system was verified by tracing them with numerical analysis, as shown in Fig. 5, where n = 32; λ = 0.0625; μ = 0.00625; the arriving group size has a discrete uniform distribution (i.e., $\lambda_1 = \lambda_2 = \lambda_3 = \ldots = \lambda_{32} = \lambda/32$); $\hat{g}$ = 16.5, and the offered load was $\hat{g}\lambda/\mu$ = 165. Under the constant offered load, a higher c implies a lower CBP. Nevertheless, the BU may not have the monotonicity under certain offered load, such as 165, as shown in Fig.5. Consequently, the optimization problem is solved by first determining c to maximize the CBP restricted by $CBP_{threshold}$, then searching for c increasingly to maximize BU.

Table 2 lists the optimal c values according to various values of offered load $\hat{g}\lambda/\mu$, and the corresponding optimized BU, where n = 16; $CBP_{threshold}$ = 4%; μ = 0.00625; λ ranges from 0.0125 to 0.575; the arriving group size was distributed geometrically (i.e., λ_1: λ_2: λ_3: … : λ_{16} = 0.18899: 0.15475: 0.12663: 0.10372: 0.08486: 0.06949: 0.05698: 0.04659: 0.03812: 0.03127: 0.02558: 0.02094: 0.01713: 0.01404: 0.01149: 0.00941), and $\hat{g}$ = 4.83682. The BU falls when c is increased to 1 more than the optimal value. Conversely, the CBP exceeds the constraint $CBP_{threshold}$ when c is 1 less than its optimal value. Thus the table shows the optimal values of c. In the OFDMA subcarrier-allocation system, the optimal c for maximizing BU can be considered to set about c subcarriers in the corresponding cell. Restated, the cell should be assigned the corresponding frequency bandwidth to support the c subcarriers.

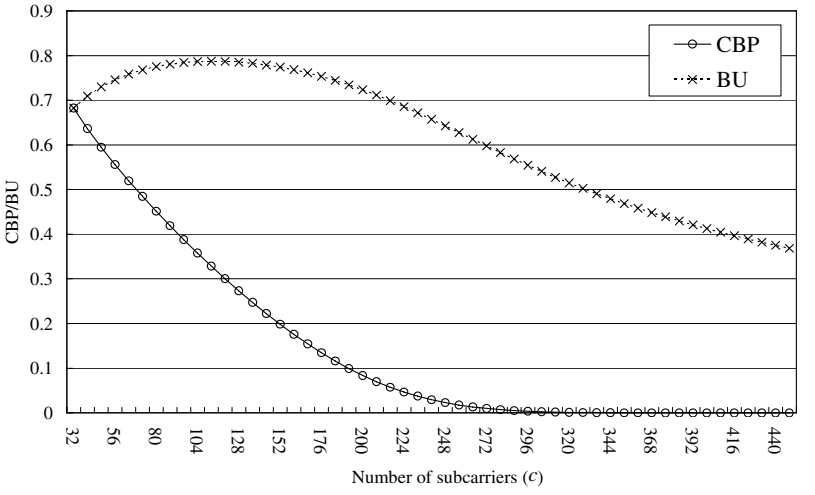


Fig. 5. CBP and BU in the OFDMA subcarrier-allocation system when the number of subcarriers c increases

Table 2. The optimal values of c for maximizing BU with CBP ≤ 4% ($CBP_{threshold} = 4\%$)

Offered load	Opt. c	CBP [c-1]	CBP [c]	CBP [c+1]	BU [c-1]	BU [c]	BU [c+1]
9.7	26	0.0451	0.0382	0.0321	0.3492	0.3412	0.3331
58.0	79	0.0402	0.0374	0.0348	0.6850	0.6802	0.6754
106.4	127	0.0408	0.0388	0.0369	0.7777	0.7747	0.7716
154.8	174	0.0409	0.0394	0.0378	0.8244	0.8222	0.8200
203.1	221	0.0402	0.0390	0.0377	0.8524	0.8507	0.8490
251.5	267	0.0405	0.0394	0.0383	0.8727	0.8714	0.8700
299.9	313	0.0404	0.0394	0.0385	0.8874	0.8863	0.8852
348.3	359	0.0402	0.0393	0.0385	0.8986	0.8977	0.8968
396.6	404	0.0408	0.0399	0.0392	0.9083	0.9075	0.9067
445.0	450	0.0404	0.0396	0.0389	0.9155	0.9148	0.9141

6 Conclusion

This paper presented the batch-arrival queueing model $M^X/M/c/c$ for evaluating the performance of the OFDMA subcarrier-allocation system. The simulation results agree with the predictions derived from the theoretical models. Additionally, the performance measures were applied to solve the subcarrier utilization-optimization problem that uses a given traffic load and a specified maximum CBP constraint, and searches for the optimal number of subcarriers to maximize the BU. As a result, the operators may utilize the results to manage their frequencies or subcarriers flexibly and to increase their profit.

References

1. Sampath, H., Talwar, S., Tellado, J., Erceg, V., Paulraj, A.: A Fourth-generation MIMO-OFDM Broadband Wireless System: Design, Performance, and Field Trial Results. IEEE Commun. Mag., Vol. 40, No. 9 (2002) 143–149
2. Intel Literature Center: Orthogonal Frequency Division Multiplexing. http://www.intel.com/ netcomms/technologies/wimax/303787.pdf (2005)
3. Jamalipour, A., Wada, T., Yamazato, T.: A Tutorial on Multiple Access Technologies for Beyond 3G Mobile Networks. IEEE Commun. Mag., Vol. 43, No. 2 (2005) 110–117
4. Dubuc, C., Starks, D., Creasy, T., Hou, Y.: A MIMO-OFDM Prototype for Next-generation Wireless WANs. IEEE Commun. Mag., Vol. 42, No. 12 (2004) 82–87
5. Zhang, Y.J., Letaief, K.B.: An Efficient Resource-allocation Scheme for Spatial Multiuser Access in MIMO/OFDM Systems. IEEE Trans. on Commun., Vol. 53, No. 1 (2005) 107–116
6. Song, P., Cai, L.: Multi-user Subcarrier Allocation with Minimum Rate Requests for Downlink OFDM Packet Transmission. Proc. of IEEE VTC'04, Vol. 4 (2004) 1920–1924
7. Bakhtiari, E., Khalaj, B.H.: A New Joint Power and Subcarrier Allocation Scheme for Multiuser OFDM Systems. Proc. of IEEE PIMRC'03, Vol. 2 (2004) 1959–1963
8. Liang, X., Zhu, J.: An Adaptive Subcarrier Allocation Algorithm for Multiuser OFDM System. Proc. of IEEE VTC'03, Vol. 3 (2003) 1502–1506
9. Kulkarni, G., Adlakha, S., Srivastava, M.: Subcarrier Allocation and Bit Loading Algorithms for OFDMA-based Wireless Networks. IEEE Trans. on Mobile Computing, Vol. 4, No. 6 (2005) 652–662
10. Han, Z., Ji, Z., Liu, K.J.R.: Fair Multiuser Channel Allocation for OFDMA Networks using Nash Bargaining Solutions and Coalitions. IEEE Trans. on Commun., Vol. 53, No. 8 (2005) 1366–1376
11. Hoymann, C.: Analysis and Performance Evaluation of the OFDM-based Metropolitan Area Network IEEE 802.16. Computer Networks, Vol. 49, No. 3 (2005) 341–363
12. Gross, J., Geerdes, H.-F., Karl, H., Wolisz, A.: Performance Analysis of Dynamic OFDMA Systems with Inband Signaling. IEEE Journal on Selected Areas in Commun., Vol. 24, No. 3 (2006) 427–436
13. Du, Z., Cheng, J., Beaulieu, N.C.: Accurate Error-rate Performance Analysis of OFDM on Frequency-selective Nakagami-m Fading Channels. IEEE Trans. on Commun., Vol. 54, No. 2 (2006) 319–328
14. Canpolat, B., Tanik, Y.: Performance Analysis of Adaptive Loading OFDM under Rayleigh Fading. IEEE Trans. on Veh. Tech., Vol. 53, No. 4 (2004) 1105–1115
15. Lee, W.C.Y.: CS-OFDMA: A New Wireless CDD Physical Layer Scheme. IEEE Commun. Mag., Vol. 43, No. 2 (2005) 74–79

Convergence Services Through NGN-CTE on the Multiple Service Provider Environments in NGN

Soong Hee Lee[1], Haeng Suk Oh[2], Dong Il Kim[3],
Hee Chang Chung[4], and Jong Hyup Lee[1]

[1] Dep't. of Information and Communications Engineering, Inje Univ., Korea
{icshlee, icjhlee}@inje.ac.kr
[2] Electronics and Telecommunications Research Institute, Korea
hsohs@etri.re.kr
[3] Dep't. of Information and Communication Engineering, Dongeui Univ., Korea
dikim@deu.ac.kr
[4] National Computerization Agency, Korea
heechang@nca.or.kr

Abstract. The NGN convergence services will play an important role to consolidate the deployment of NGN at the initial phase. At the initial stage, scenario-based approach is expected to act as an efficient way to pre-design before the actual implementation of NGN, the subject for providing the convergence services to the customers. This paper first proposes a deployment model of the NGN convergence services. In this model, NGN-Service Providers (NGN-SP) interact with Service Coordination Function of NGN Service Stratum whereas NGN convergence service terminal equipments (NGN-CTE) provide convergence services to the end users after procedures in End User Service Composition Function. A service composition scenario for the convergence services will be introduced, using the model, to show the usefulness of the proposed deployment model to construct new convergence services on this scenario-based approach.

Keywords: NGN, NGN-SP, NGN-CTE, convergence.

1 Introduction

Concepts of Convergence Service have been regarded as one of most important issues for NGN deployment. The standardization activities of ITU-T have mainly focused on the requirements and frameworks around the convergence services [1]. The scenario based approach in ITU-T has accelerated their concentrations on this issue. Even with these efforts, the actual deployment of convergence services look too far to be realized.

On the other hand, there has been a trend of terminal devices with various capabilities and interfaces. This trend seems to choke the end users' desires and preference on these various services and vice versa. However, terminals with functions satisfying this trend might require larger capacities and accordingly very high costs, which lead to make the deployment in the practical market even harder. A

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 422 – 430, 2006.
© Springer-Verlag Berlin Heidelberg 2006

possible solution on this problem is to allot the functions between terminal side and network side. This way also provides a breakthrough to the standardization efforts of ITU-T on the Convergence Services in that the allotted functions in the network side can be provided using the standardized Convergence Service scenarios.

The allotment of the service functions mainly occurs in the NGN-CTE (NGN convergence service terminal equipment) and NGN Service Stratum. The NGN-CTE will play an important role to consolidate the deployment of NGN at the initial phase. At the initial stage, convergence may be initiated at the terminal side with less burden of the network or service provider side. Thus the deployment model for NGN convergence service terminals will be explored in this paper. Next, the way to compose a Convergence Service using this deployment model will be explained using the form of scenario.

2 Convergence Services in NGN

According to the ITU-T Rec. Y.2011 [2], Convergence Services is defined as the ability to deliver a wide variety of services including voice, video, audio and visual, data, via session and interactive based services in unicast, multicast and broadcast modes.

Convergence Services can be provided not in a single way. A terminal having multiple media interfaces and powerful processing can handle some services that are very similar to the convergence services that looked as only possible from the network side. But this may bring complex, expensive, and heavy terminal equipments to the end users, which will finally prevent the smooth migration to NGN.

As results of this convergence, Service Profile Management in the network side can be feasible.

Users' Benefits from the Service Profile Mgmt. in the network side are as follows:

- Maintaining the Service Profile that includes Frequently-visited sites or Preferred-services;
- Maintaining personal preference of End Users;
- Automatic Parental Control/Contents Filtering;
- Saving the costs for the Terminals of End Users;
- Finally providing more conveniences to End Users (Lighter and Cheaper Terminal Devices can be provided).

On the other hand, Providers' Benefits from the Service Profile Mgmt. in the network side are as follows:

- Authentication/Authorization in the network;
- Manageability of Per-Usage Charging;
- Flexible service creation based on the End Users' Needs;
- Getting more chances for profits from the new businesses.

3 NGN-CTE and NGN-SP

3.1 Balance Between User and Network Sides for Convergence Service Provision

Convergence services in NGN can be provided without changes in the network side if multi-functional terminal can support all kinds of convergence services end users demands. However, this approach is not regarded as practical on the cost and business aspect of the convergence services. On the contrary, the convergence services can be actuated using the capabilities inside NGN Service Stratum while using the low-cost terminals at the user side.

These aspects require a kind of guideline to both the user side and the network side. The former can be specifically embodied as the NGN-CTE while the latter as the NGN supporting multiple NGN-SPs (NGN Service Providers).

3.2 Aspect of NGN-CTE

First, we will consider the requirements for the NGN-CTE as follows:

- Memory with limited size for maintaining cache for storing the previous service creation information;
- Various media interfaces;
- Display screen (VoD or DMB can be provided, projection can a good choice);
- Cheaper and lighter;
- Voice as mandatory (e.g. emergency communication purposes).

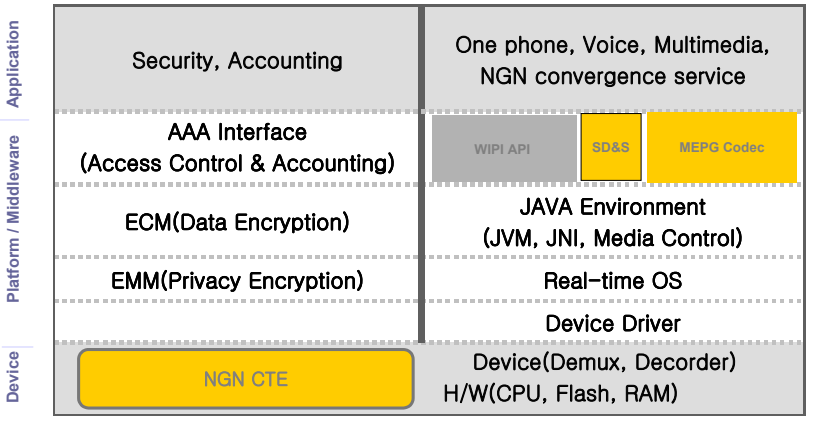


Fig. 1. Logical structure of NGN-CTE

Figure 1 shows the logical structure of the NGN-CTE. This structure accords to the ISO/OSI layering convention.

The top layer of this structure includes:

- Service offering intended by the Service Provider;
- Functions and capabilities provided by CTE;
- Security and accounting.

This layer consists of programs, information about programs, IP address; in short, the essential items needed to enable NGN convergence services over NGN. The middle layer of this structure includes:

- Services and functions:
  - Wireless network interfaces provided by WIPI API;
  - Information for service discovery and selection service according to the SD&S protocol;
  - Audio and video streams are multiplexed into a valid MPEG-2 transport stream.
- Network access and accounting:
  - AAA interface in order to provide network access and accounting.
- JAVA environment:
  - JVM, JNI and Media control are executed in JAVA environment.
- Contents encryption:
  - Contents data encrypted and scrambled by ECM encoder as shown Figure 3, where ECM means the encrypted contents.
- Privacy encryption:
  - Privacy data encrypted EMM encryptor as shown Figure 3, where EMM means the encrypted privacy data.
- Operating System:
  - NGN CTE is executed by real time OS.

The bottom layer of this structure includes:

- Devices:
  - Audio and video streams are multiplexed/demultiplexed and encoded/ decoded by the MPEG encoder/decoder.
- Hardware:
  - H/W of NGN-CTE is composed of CPU, flash memory, and RAM, etc.

3.3 Aspect of NGN-SP

Multi-provider environments assume several different situations such as service providers only inside of NGN, service providers in NGN and other networks, and service providers in different networks each. According to the FRA document of ITU-T [3], NGN-SPs are classified as Retailing Service Provider, Wholesale Service Provider, and Value-Added Service Provider.

Each service provider has its role as follows:

- *Retailing Service Provider*: The role that has overall responsibility for the provision of a service or set of services to users associated with a subscription as a result of commercial agreements established with the users (i.e., subscription relationships). The user profile is maintained by the retailing service provider.

Service provision is the result of combining wholesale network services and service provider service capabilities.

- *Wholesale Service Provider*: The role that combines a retailing service provider's service capabilities with its own network service capabilities to enable users to obtain services.
- *Value-Added Service Provider*: The role that provides services other than basic telecommunications service (e.g., content provision or information services) for which additional charges may be incurred. These may be billed via the customer's service provider or directly to the customer.

This implies that it is needed to clarify the relation between these service providers to construct the effective convergence services and differentiate each service in the perspective of service categorization. It looks evident that Transport Providers and Service Control Providers require more study to be included in the category of Service Providers.

4 Deployment Model for the NGN Convergence Service

It is clearly needed to prepare a deployment model for composing a convergence service scenario before the actual deployment. Hence, a deployment model consisting of parts dealing with terminal side and service provider side. The former is called End User Service Composition Function while the latter is called Service Coordination Function. Figure 2 shows the proposed deployment model for the NGN convergence

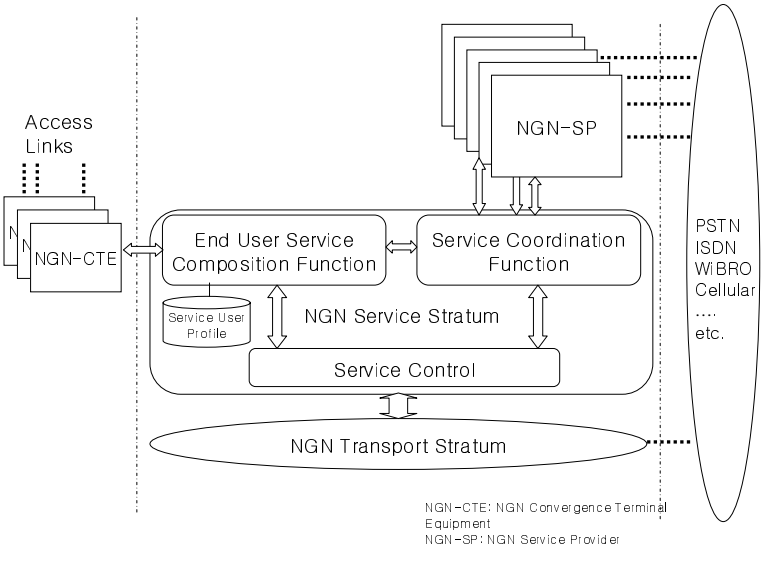


Fig. 2. Deployment model of the Convergence Services over NGN

service. NGN-Service Providers (NGN-SPs) interact with Service Coordination Function of NGN Service Stratum whereas NGN convergence service terminals (NGN-CTE) provide convergence services to end users after AAA procedures. Various types of NGN-SP may provide services that will be used as components of the convergence services in NGN Service Stratum.

5 Convergence Service Scenario

5.1 Scenario of End User Service Composition

Using the deployment model shown in Figure 2, an end user service can be composed according to the procedure as shown in Figure 3. The detailed procedure is as follows:

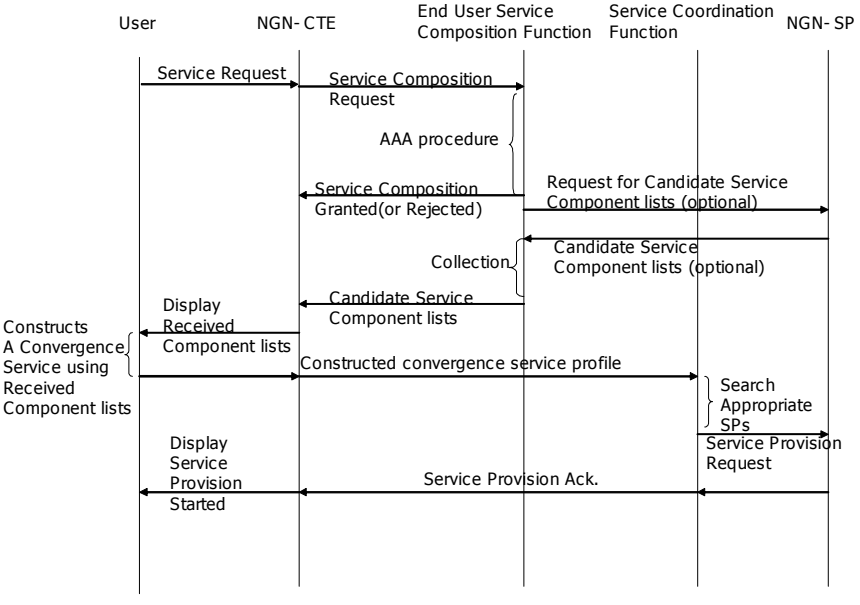


Fig. 3. Procedure of the Convergence Service Composition

- A user requests a new service composition using his NGN-CTE.
- NGN-CTE sends the service composition request including the identification information to the End User Service Composition Function.
- After receiving the service composition request, the End User Service Composition Function identifies the requesting user via AAA procedure using the pre-stored service user profile.
- If the AAA procedure is completed, End User Service Composition Function sends the candidate service components to the NGN-CTE.
- NGN-CTE displays the received components on the screen.

- The user constructs a convergence service by selecting required components.
- NGN-CTE sends the constructed convergence service profile to the End User Service Composition Function.
- End User Service Composition Function relays the received profile to the Service Coordination Function.
- Service Coordination Function then starts to search appropriate NGN-SPs.
- If appropriate NGN-SPs are found, Service Coordination Function sends the service provision requests to the correspondent NGN-SPs.
- After receiving the service provision request, the correspondent NGN-SPs start to provide the services with the simultaneous sending of the Acknowledgement to the Service Coordination Function.
- Service Coordination Function relays the Acknowledgement to the NGN-CTE and End User Service Composition Function.
- The user accepts the Acknowledgement from NGN-SPs and is provided the requested convergence service.

5.2 Updates of the Service User Profile

Considerations on the Service User Profile are essential for the Convergence Services to be successfully deployed in NGN. Continuous updates of the Service User Profile are essential for the successful composition of the Convergence Services. To update the profile effectively, the network-friendly repository tool, e.g. LDAP (Light -weight Directory Access Protocol), is more appropriate. The update of the Service User Profile can be actualized based on LDAP as shown in Figure 4. The LDAP server dealing with the Service User Profile maintains the Service User Profile Information

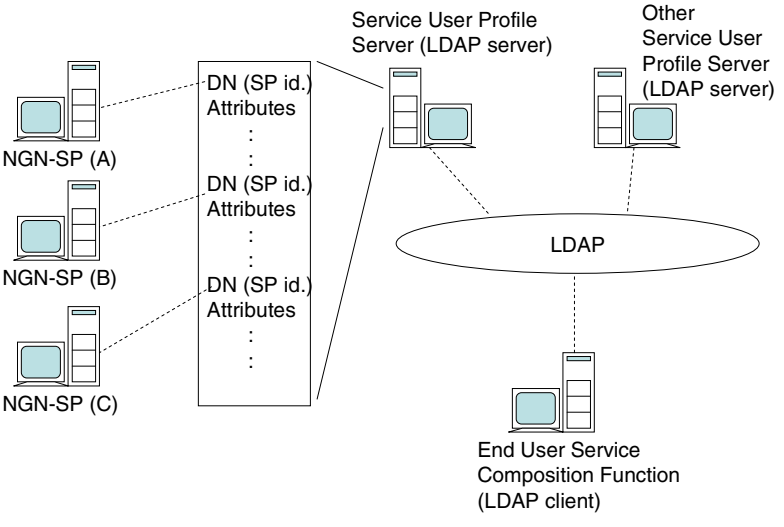


Fig. 4. LDAP-based updates of the Service User Profile

Base based on the input from the participating NGN-SPs. Participating NGN-SP can be a station of LANs or other types of networks while each NGN-SP has its own IP address. Table 1 shows an example of the Service User Profile shown in Fig. 4.

Table 1. An example structure of the Service User Profile

		User 1		User2		….
User Profile	Name	James Smith		David Miller		….
	Email address	james@whois.com		david@any.com		….
	Phone number	010-234-5678		023-342-1090		….
	Other attributes :	: :		: :		….
Previously Used Service Profile 1	Connected User Address	IP	203.146.234.5	IP	127.202.189.235	….
		MAC	5A:2B:23:0D:1A:2A	MAC	61:2C:67:95:3A:1F	
	Participating SP Address	IP	201.35.23.52	IP	201.35.23.52	….
		MAC	1D:1B:43:25:3C:21	MAC	70:2B:1C:7F:E0:37	
	Participating SP Address	IP	215.129.230.36	IP	131.32.21.49	….
		MAC	NA	MAC	NA	
	:	:		:		:
	Service Type	Voice+Video		Voice+Data		….
	Service Start Time	2006:02:23:02:27		2006:02:19:13:11		….
	Service End Time	2006:02:23:02:29		2006:02:19:13:14		….
	:	:		:		….
	:	:		:		….
Previously Used Service Profile 2	Connected User Address	IP	201.132.3.12	IP	202.121.34.2	….
		MAC	4D:23:12:3F:2A:7D	MAC	3E:F4:12:35:6B:F6	
	Participating SP Address	IP	201.35.23.52	IP	203.125.23.78	….
		MAC	NA	MAC	5D:1B:43:25:3C:69	
	Participating SP Address	IP	203.32.56.21	IP	201.34.28.121	….
		MAC	NA	MAC	NA	
	:	:		:		:
	Service Type	Data+Video		Voice+Data+Video		….
	Service Start Time	2006:02:22:22:31		2006:02:01:16:12		….
	Service End Time	2006:02:22:23:39		2006:02:01:17:02		….
	:	:		:		….
:	:	:		:		:

5.3 Service Coordination Among NGN-SPs

Convergence services can be composed of various service components from different NGN-SPs. Without some mediation between NGN-SPs such as service coordination, the Convergence Services can hardly be provided. Service Coordination Function in

the proposed deployment model seeks the appropriate NGN-SPs among candidates after receiving Convergence Service Request from the end user. The list of NGN-SP must be maintained for this search process. Pre-registration of NGN-SPs into the list is also needed.

5.4 Customization and Personalization of End User Services

NGN-CTE can request the convergence services customized to the end users preference. The preference of end users includes the selection of NGN-SPs that provide same services at the lower cost than the competitive ones. Customization and Personalization of the service provision can be easily approached using a kind of GUI at the NGN-CTE. The detailed aspect of the GUI is for further study.

6 Conclusion

A deployment model and a service composition scenario are proposed for providing the Convergence Services to the end users. The scenario shows that the proposed deployment model can be useful for creating new Convergence Services over NGN. In addition, the LDAP-based procedure for updates of Service User Profile is presented. Customization and Personalization of service provision is expected to be studied more for completeness of this approach. More study is also needed to clarify the details to this brief outline of the Convergence Services in NGN.

Acknowledgments. This work was supported by the Inje Research and Scholarship Foundation in 2004.

References

1. ITU-T FGNGN-OD-00248R1 (2005), Revised working draft TR-CSF Version 3.0 (NGN Release 2)
2. ITU-T Recommendation Y.2011 (2004), General principles and general reference Model for NGNs
3. ITU-T FGNGN-OD-00223, Draft FGNGN-FRA Version 6.3.

Proposal of Operation Method for Application Servers on NGN Using Unified Management Environment

Atsushi Yoshida, Yu Miyoshi, and Yoshihiro Otsuka

NTT Network Service Systems Laboratories, NTT Corporation,
9-11, Midori-Cho 3-Chome, Musashino-Shi,
Tokyo 180-8585, Japan
{yoshida.atsushi, miyoshi.yu, otsuka.yoshihiro}@lab.ntt.co.jp

Abstract. The Next Generation Network (NGN) has a multilayer logical construction. Network carriers use various tools in complex procedures to operate application servers on the NGN. We try to establish a method for operating application servers in a unified management environment. This method operates between the network layer, which manages applications. First, we select requirements to achieve a unified management environment for application server operation. Then, we select a test function and a monitoring function from operation functions, in an experiment to confirm the achievability of our method. We test the achievability of two application server functions, which supply web services in the experiments. Our method supplies these functions without programming and manual procedures.

1 Introduction

1.1 Next Generation Network

Recently, a lot of network carriers such as British Telecom (BT) and Nippon Telegraph and Telephone Corporation (NTT) have been aiming at the introduction of NGN, which resolves the problems of existing public switched telephone networks (PSTN) and enables various services such as multicasting multimedia content between interest-sharing communities over internet protocol (IP) based networks. To achieve the NGN successfully, such carriers are discussing the standardization of NGN architecture at conferences such as International Telecommunication Union Telecommunication Standardization sector (ITU-T), European Telecommunications Standards Institute (ETSI), and Telecom and Internet-converged Services and Protocols for Advanced Networks (TISPAN) [1][2][3]. Researchers at TISPAN define the NGN architecture (Fig. 1) as a layered structure with an IP-based transport layer and a service layer.

In detail, they define IP-connectivity assurance systems in the transport layer and service provision components in the service layer. The group of service provision components for business and end users is in the service layer of an NGN-based infrastructure. For example, Core IP Multimadia System (Core IMS) and

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 431–440, 2006.
© Springer-Verlag Berlin Heidelberg 2006

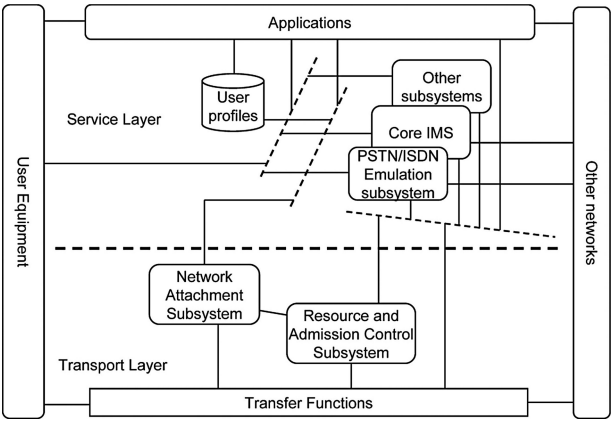


Fig. 1. TISPAN NGN Architecture [2]

PSTN/ISDN (integrated services digital network) emulation subsystems are service provision components. By using service provision components on the service layer, telecom companies can develop a business infrastructure to provide services such as fixed mobile convergence (FMC).

There are few service provision components on PSTN, and network carriers do not have much know-how about service provision technologies. To achieve the NGN, network carriers need to obtain these technologies. Network carriers need to research and develop service provision components for the NGN.

1.2 New Generation Operations Support System

The NGN has a complex architecture to provide high service provision performance. Services on the NGN are provided by combining several service provision components that work on NGN servers. For providing seamless services to NGN users, network carriers should manage each component properly; thus, the operation and management tasks are difficult. This complexity causes high NGN maintenance costs for network carriers. We need to reduce the cost of achieving the NGN.

Researchers on TeleManagement Forum (TMF) standardize network operation processes to reduce the cost of operating the NGN. This standard is NGOSS [4]. NGOSS has four frameworks.

eTOM Business process framework
SID Enterprisewide information framework
TNA System Integration framework
TAM Applications framework

The researchers in the TMF define enhanced Telecomm Operation Map (eTOM) and classify operations and business processes in the telecom domain

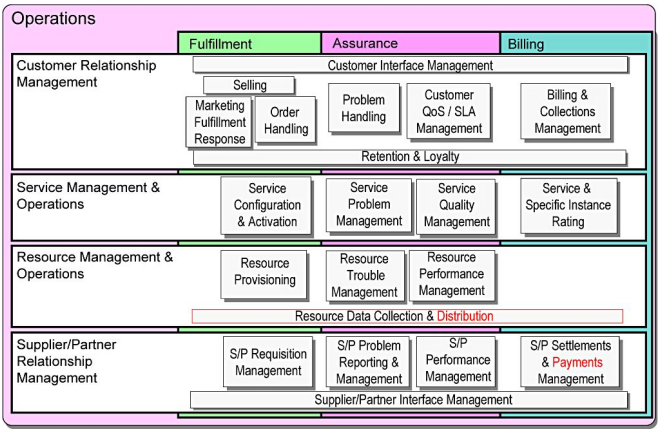


Fig. 2. eTOM Operations Processes [5]

(Fig. 2) [5]. By using eTOM, we can map between business processes and operations support system (OSS) functions.

Symbolized by frameworks such as eTOM, operation software tools are classified by operation functions such as those of the service management layer and the resource management layer. Operators in network carriers need to use different tools in complex procedures if the management objects are in different layers, even when the same services are operated. To integrate operational tasks is difficult for operators. For example, they need to use IP reachability check tools such as ping and TCP/UDP state check tools such as netstat for managing a server.

For resolving the above problems, we propose the examination of the unified operation method, which examines service provision processes of servers on the NGN service layer. Our purpose is to improve operator usability. We propose a unified operation method for application servers on NGN.

In section 2, we introduce design of an automatic management technique for server components such as hardware, OS, and applications. The design of the proposed method is described in section 3. We outline our verification of our proposal in section 4. After that, we examine the achievability of the proposed method in section 5 and show the experimentl results in section 6. And we discuss the results and achievability of our proposal in section 7. In section 8, we conclude our work.

2 Related Work

To reduce server management costs, there is design of an automatic management technique for server components such as hardware, OS, and applications. This technique is called ISAC [6], and it provides automatic server management by a PCI-X card-based system that plugs into an available slot on the server. According to user-defined management scenarios, a PCI-X card and software installed

on the server manage the server components and communicate with each other to enable automatic management. A comparison between the proposed method and ISAC is shown in Table 1. A purpose of ISAC is an independence of management method from a server condition. It doesn't consider operator usability using the unified environment.

Table 1. Comparison between Proposal Method and ISAC

	automatic recovery	realtime response	network element management	unified environment to operator
ISAC	Y	Y	N	N
proposal method	future works	N	Y	Y

3 Proposal of Application Server Test Functions

In this paper, we propose a method for managing servers that are connected in NGN infrastructures. Purposes of our method are a smooth migration of operation techniques from existing operation techniques for network management and a reduction of the impact of the introduction of the NGN on the operation cost.

First, this section describes a unified server management method for IP transport functions and applications. Next, we explain the configuration support technique of this proposal. We have developed a technique for operating network equipments (NEs).

3.1 Application Server Test Functions

The service provision components of the servers on NGN often use message passing technique. This technique is used, Universal Description, Discovery, and Integration (UDDI), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), which is used to achieve web services, and Session Initiation Protocol (SIP), which is used to achieve voice over IP (VoIP) services. In each technology, messages are encapsulated by the implemented library of the technique, so the content of these messages cannot be read by users. Although this encapsulation is necessary to improve productivity for users and developers of the services, it also makes the monitoring and controlling of services difficult for network carriers who must develop a service platform and assure the quality of services. That is, in this situation, network carriers need to use multiple tools because they must display and analyze text messages by using dedicated monitoring tools.

Hence, unifying the management environment is important for a smooth workflow and efficient management operation. Therefore, we propose a method that achieves unified monitoring and control of services by using text messages and software that executes text manipulations (Fig. 3).

This method is characterized by the display of the content of the message in readable form and supporting the control of exchanging messages. These features achieve user-friendly management of applications for operators. That is, our

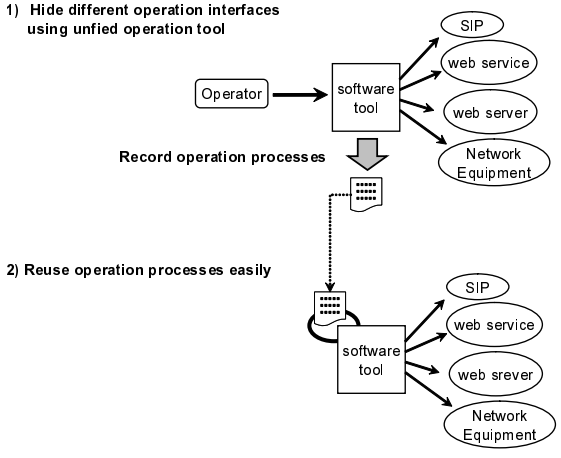


Fig. 3. Unified Server Management Method

method uses the command line interface (CLI) based supprt technique that is used in operations such as the configuration or validation of NEs.

This enables operators to execute service operations using the same manipulation as that for NE operations and to unify the operation environment. This unified operation environment achieves IP network management in servers and control by a single software tool. In addition, this control brings about the following effects. For service developers, promoting efficiency such as that achieved by automation becomes easy. Moreover, the scope of the automation is expanded. For operators, there is no longer a need to develop and manipulate multiple tools and specific clients. As a result, NGN operations become efficient.

3.2 Configuration Support Technique

Presently, we are developing a technique that supports the configuration of NEs like routers or switches. This technique reduces the operation load by retaining operator know how and sophisticated operations like configuration or validation (Fig. 4) [7].

Although there are policy-based configurations, commercial off-the-shelf (COTS) products with convenient GUIs, and other operation techniques that have advanced functions, traditional CLI processes continue to prevail in configuration management operations such as the validation of network equipment, configuration, backup, and log collection. Several reasons have been advanced for this. For example, we considered that established functions cannot be maintained at the speed at which network specifications are updated. Another key reason is that a great deal of operator knowledge and experience is stored in conventional processes on the CLI. Then, our configuration support technique attempts to retain and provide a more sophisticated manipulation method on the CLI, which is familiar to operators to reduce the operation load. The software tool developed based on this technique records the process that an operator

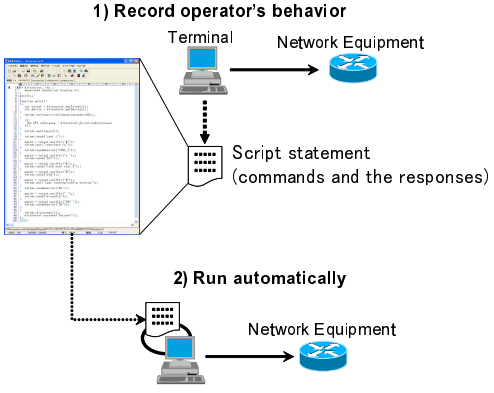


Fig. 4. Configuration Support Tool

has executed in a script format, replays the recorded process, and executes that process on a schedule determined by a timer. Therefore, once a manipulation is performed on the CLI, operators can automatically execute that manipulation that has already been performed. This significantly reduces the operation load of routine work. In addition, our technique focuses on the script format. By defining the script in a operator-friendly format, flexibly modifying the commands, the conditions, and the parameters recorded in the script becomes easy. This enables operators to conduct tests while reviewing and modifying the test entries on site. In particular, during the validation, the time lag caused by software programming development is reduced.

4 Outline of Verification

We executed an experiment to verify the realizability of the proposal method. We chose components of web services provided on the application server, as the subjects of this verification.

In general, a web service consists of application servers that provide a web service, a web site that provides services composed of web services, and a client that uses web services.

An application server that provides web services offers a function to send WSDL messages and a function to send and receive SOAP messages to provide web services. To verify the application server, monitoring and controlling these two functions are required. Web services use several resources, so when a failure of a web service occurs, we need to check the statuses of the resources used by this service and specify the causes of the failure. The statuses of these items need to be checked to verify the application server.

1. Reachability of IP packet
2. HTTP daemons
3. Platforms used to provide Web Service (e.g., tomcat)

4. WSDLs
5. Web services

We need to use a different tool to check each item shown above to verify the application server by existing verification methods. Some points need to be checked manually, so we think there are some processes that need to be changed to automatic execution.

5 Experiment

Here, we explain the procedure of verifying an application server that provides web services. First, we check the reachability of an IP packet by sending a ping command to the application server. Next, we check whether HTTP daemons on the application server are running by sending specified HTTP requests to the application server, which platform is being used to provide a web service on the application server, and whether WSDLs, which describe web services on the application server, can be obtained. We check whether the syntax of the obtained WSDLs is correct. Therefore, if the syntax is correct, we can extract the operation methods from each WSDL. We execute operations in a particular WSDL in the correct order, and check each response of the operations. Once the obtained WSDLs are checked appropriately by the above process, we store the XML messages used by the check (the obtained WSDLs, request messages, and response messages) as script files. By partially rewriting these scripts and resending them, we can automatically check a particular application server. In this paper, we experimented with an application server to verify the procedure described above.

In the experiment, we use telnet software to obtain WSDL text and to request web service operations. To check the function to obtain WSDL texts, we capture HTTP requests generated by a web browser, and we resend these requests using telnet software. We verify the realizability of checking the function to obtain WSDL texts (Fig. 5) using this operation.

We capture SOAP messages generated by dedicated client software to check the function to execute web services, and we resend these messages using telnet software. By this operation, we verify the realizability of checking the function that executes web services (Fig. 6).

In these experiments on web service and web service tests, we can achieve text processing of messages by partially rewriting SOAP messages, which is carried out by partially changing the SOAP message parameters. We verified these two application server functions, to supply WSDL and to execute web services, with the following check items. Thereby, the availability of the proposed method was estimated.

- check item 1: Can we monitor these functions? (Monitoring)
- check item 2: Can we experiment various tests? (Experiment)
- check item 3: Can we execute the automation of these two functions? (Automation)

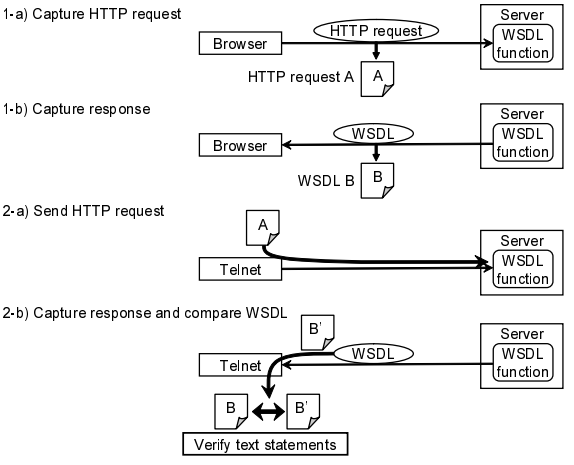


Fig. 5. Verification Procedure of WSDL

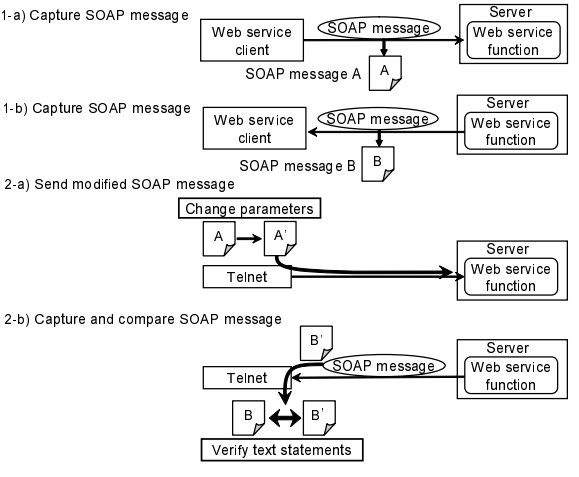


Fig. 6. Verification Procedure of Web Service

We explain these check items below.

Monitoring: Verify whether the monitoring of the application server can be performed in a general environment, without using a dedicated monitoring software.

Experiment: Verify whether the application server experiment can be performed in a general environment, without using any dedicated monitoring software. The experiment and checking the web service can be executed by changing parameters of request messages, which lead to constructing a unified environment of verifying application servers. From the results of the experiment, con-

sider whether operators can execute text processing of request messages, and whether the experiment can be executed automatically.

Automation: Estimate the possibility of automating the above monitoring and experimental functions from the results of the experiment. In addition, we consider requirements for the automation.

6 Experimental Results

Results of the experiment are shown in Table 2. In the first step, we performed a WSDL function test. We sent a sequence of captured HTTP requests that were generated by a browser software to call an application server. In this test, we confirmed WSDL functions on application servers.

Table 2. Experimental Results

	resend	confirm result
experiment of WSDL function test	Y	Y
experiment of web service function test	Y	Y

In the second step, we checked the service provision capability using a test tool. We used a dedicated web service client software and captured transactions of SOAP messages between this client and the server. Next, we sent captured request messages to the server, and captured correct responses from the server. In this trial, we confirmed that there was no need for setting environment variables or compiling client software compared with using a Java client. Therefore, the capability of constructing a more generic testbed for the application server was demonstrated. In addition, many packets were generated in one request, but there was no effect on the response of the server. We edited various SOAP message parameters contained in XML tag containers and HTTP headers. Then we confirmed that we could use a modified request.

7 Discussion

Periodic WSDL monitoring is useful because HTTP requests are normally used. Resending a request with a different parameter is not required, so WSDL is not changed until the alteration of service specification. We consider two WSDL test capabilities can be automated by script using specified URL of WSDL. In this way, we can use retrieved WSDL for remote monitoring functions and automated checking of functions of web services in test scripts.

On the other hand, we can make a function for monitoring web services using captured requests and responses by normal client and the server. Increasing the efficiency of building a set of correct SOAP messages for beginning the test is important. In the experiment on service level transactions, we confirmed that rewriting HTTP headers and parameters is easy; therefore, testing web

Table 3. Results of Checked Items in the Experiment

	check item 1: Monitoring	check item 2: Experiment	check item 3: Automation
WSDL	Y	-	Y
Web service	Y	Y	Y

service messages using automatic parameter rewriting with text processing tools is feasible. HTTP header translation is simple and a parameter type is required in the SOAP message. In these web service test functions, the operator must define the range of parameters before the test, and the software testing tool will make various parameters automatically. We believe that monitoring and testing web services using the same method as that of monitoring of network elements is achievable (Table 3).

8 Conclusion

We defined a method of verifying web services. With this method, we were able to reduce the work and amount of time to perform tests of web services arranged in an IP network. Test functions improved web service tests as listed below.

- This method makes a unified operating environment for communication functions and service functions using text processing.
- Automatic operation is easily performed.
- There is no need to develop dedicated software tools.

In conclusion, we consider that we have achieved a test function with automatic modification of parameters.

References

1. A. Kurokawa and I. Azuma: The NGN standardization trend in ETSI TISPAN (In Japanese) NTT Technical Jornal Vol. 18, No. 4, (2006)
2. ETSI ES 282 007: Telecommunications and Internet Converged Services and Protocols for Advanced Networks (TISPAN) IP Multimedia Subsystem Functional Architecture. (2005)
3. ITU-T Y.2001: General overview of NGN.
4. TM Forum: New Generation Operational Support System (NGOSS) PART 1 (Business Case), TMF051, (2001)
5. TM Forum: Enhanced Telecom Operations Map (eTOM) The Business Process Framework. Addendum D, Process Decompositions and Descriptions Release 5.0, GB921D, (2005)
6. T. White, D. Carvert, and J. Litkey: Design of an Autonomic Element for Server Management. Proceedings of the Second International Conference on Autonomic Computing (ICAC'05) (2005)
7. Y. Miyoshi and T. Kimura: Interface Blending/Diagnosis Technology for Automatically Generating an Interface Adapter. NTT Technical Review. Vol. 3. No. 10. (2005)

IP/WDM Optical Network Testbed: Design and Implementation

H.A.F. Crispim, Eduardo T.L. Pastor, Anderson C.A. Nascimento, H. Abdalla Jr, and A.J.M. Soares

Electrical Engineering Department, University of Brasília – Brasília-DF, CEP 70910-970, Brazil
hcrispim@gmail.com, eduardo@labcom.unb.br, {andclay, abdalla, martins}@ene.unb.br

Abstract. This work presents the design and implementation of an optical transparent IP/WDM network testbed. The implemented software allows the characterization of the transport, control and management planes of the network. Furthermore, it was developed a graphic user network interface for the client/management relation in the optical network. We adopted a centralized the control and management planes, obtaining economy of resources and time performance. The transport plane physical model allows the simulation of transponders, amplifiers and OXCs. They are connected with the control and management plane through the use of UDP/IP protocol and using XML for supporting the information base. The control plane provides routing and protection in the event of data plane failures. The management plane has a functional structure based on fail management, on the network configuration program, on performance monitoring, on log of changes and on the access security system.

Keywords: IP/WDM Optical Network Testbed, software simulation, Control, Management and Transport Plane, WDM, Web Services.

1 Introduction

The ever increasing demand for bandwidth, flexibility and reliability has led to the proposal and implementation of new network architectures. Due to its potential advantages, optical networks have been considered in situations where there is need of high bandwidth and low latency communication. Among many proposals, IP-centric control plane within WDM optical networks is a widely considered solution for dynamic provisioning/restoration of lightpaths [1].

Several initiatives to build WDM backbones are under development [2-3]. Passive Optical Network (PON), next generation SONET/SDH generic framing procedure (GFP) and Ethernet Resilient Packet Ring (RPR) are helping further streamline vertical "data optical integration". Given this evolution, there is a growing need for testbed development in order to provide much-needed "proved-in" value [4].

The physical layer of optical networks is composed of photonic devices, which are rather expensive. Thus, the high costs of the implementation of this kind of network

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 441 – 450, 2006.
© Springer-Verlag Berlin Heidelberg 2006

difficult the research in this field. A possible way to overcome this problem is the development of simulation scenarios, tools and techniques. This simulations, however, requires a careful design and study as to achieve a realistic scenario, where details of the planes and their elements are well represented.

In this paper we describe the design and implementation of a transparent IP/WDM Optical Network Testbed. The implemented testbed (designated hereon as LabCom testbed) considers the control plane and the management plane in a centralized model. For the management plane the development of a friendly and intuitive web-based O-UNI (Optical – User Network Interface) is presented. Our system provides the network manager with management and security mechanisms. The network is presented in figure 1.

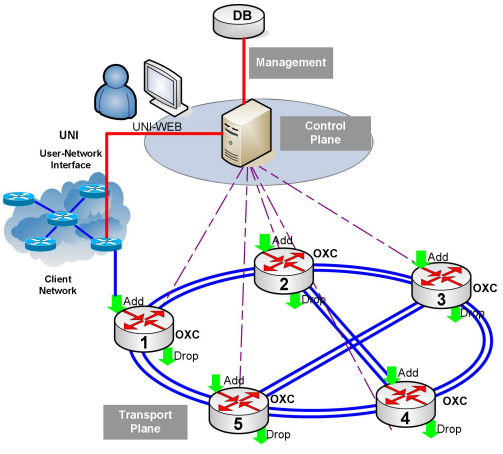


Fig. 1. Complete model of this work

2 Network's Architecture

2.1 Physical Environment

We show our built scenario in figure 2. In this proposed architecture, the transport network is formed by five nodes. Each node is connected in one Ethernet switch by a VLAN connection, forming five VLAN connections. The control plane and management functions are implemented in a personal computer (PC), forming a centralized network architecture.

The logical topology is established by VLANs. Each VLAN corresponds to an optical node link (OXC) of the transport plane. The sixth link connects the transport plane to the control plane. This ensures the isolation of the broadcast domains and also, it limits the transmission rate for each VLAN. We realized tests with ports working at a rate of 1 Mbps.

Each optical node is simulated by a IBM PC PIII with 128 MB of RAM, processor of 700 MHz and a 10/100 Mbps Ethernet network card connected to a switch. The

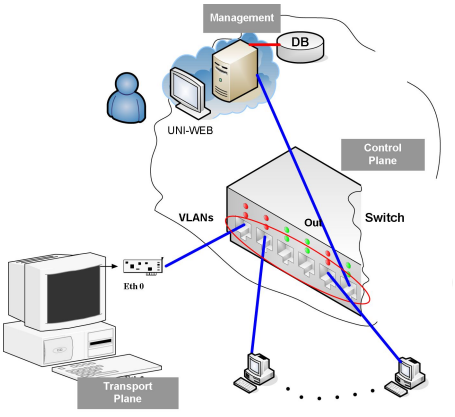


Fig. 2. Network's Physic architecture

computer which executes the control and management system is a HP PIV 1.8 GHz with 256 MB of RAM and a 10/100 Mbps Ethernet network card connected to a switch.

2.2 Management Plane

The management plane works based on fail management. The management plane consists of the network configuration program, the performance monitoring, the log of changes and the access security system. The System Management Application Entity (SMAE) is separated in two parts. Those parts are implemented by different processes (in the application layer), situated at the management unit and at the agent unit [5]. Thus, we have a centralized management process. The management objects (OXCs. Amplifiers and Transponders) register their parameters in their own information management base that will be distributed among the managing entity and the agent in each optical element. The information which corresponds to the physical elements read status is transmitted in the XML format.

Optical User Network Interface (O-UNI Web). The routes can be requested graphically by means of the UNI-Web interface. Therefore, the computer hosting the management functionalities in our LabCom Testbed can be accessed by HTTP. The proposed UNI-Web based interface [6] facilitates the interaction of the user with the transport network, allowing an efficient and friendly operation. Another functionality of our graphical interface is to make the auditing process completely transparent to the network manager.

UNI-Web can create or eliminate optical links, get information on the network and nodes state, of ports and switches, as well as visualizing the occupation of the wavelengths in each fiber. It can also provide the network manager with information on which requests were not taken care of and other administrative information. He can also audit the users and executed commands, among others functionalities. Figure 3 shows the screen of the UNI-Web.

The graphic environment was developed in web model using Java Server Pages – JSP technology. In this interface, the user is authenticated by means of a password. The system monitors users actions through auditing mechanisms made in a relational model and implemented in PostgreSQL V8.1.2.

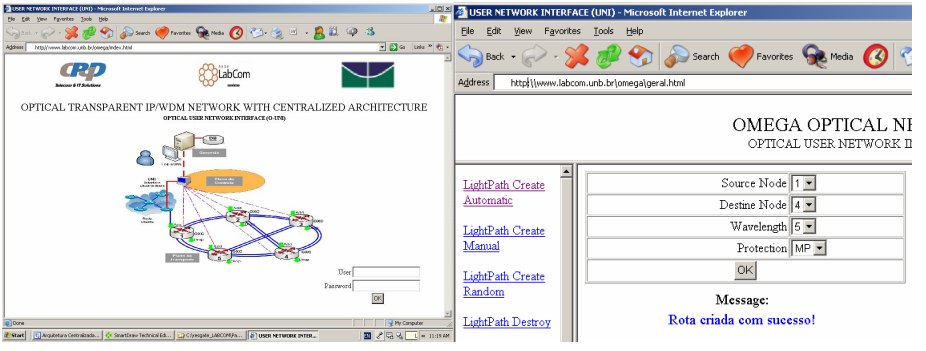


Fig. 3. Screen of the UNI-Web

2.3 Control Plane

The control plane, as explained before, was implemented in a centralized fashion. Its main function is to reroute failed connections in the event of data plane failures [7]. We defined a UNI interface as an overlay model. The interaction between the transport and control plane follows the client/server model.

For security reasons and reliability, the control system permits only one TCP/IP connection with the management system.

The communication between the control system and the agents implemented in the optical devices occurs over UDP/IP connections, since in our application the error rates will be negligible. This system is implemented based in three threads, each one with the following functions: a) to connect to the management system and implement the business rules; b) to manage the network link stability by means of implementing a Hello protocol; and c) to communicate with various nodes compounding the physical network as to manage actions such as sets, gets e traps operations.

The controlling is performed in the following logical order: a) initially the network's file configuration is read, it describes the physical topology and specifies the physical links between the optical devices; b) the file topology content is parsed, in order to evidence errors; c) the shortest paths between network elements are computed by using Dijkstra algorithm; d) alternative and disjoint paths are computed. Those paths are then made ready to substitute the original paths in case of link faults; e) the controlling mechanism (working in standby) is then ready to receive the lightpath allocation and deletion requests.

The whole control system was developed in C++ KDE environment on Linux system operational - Fedora.

2.4 Transport Plane

The physical layer elements were simulated using a graphic environment in Java. We implemented three devices: the optical cross-connects (OXCs), the amplifiers and the transponders. Each optical node is composed by: an OXC capably to commute eight lambdas from ITU grid, eight transponders to adapt the signal wavelength to the ones used in the network and eight amplifiers. Specific optical nodes can be simpler than the structure we just described, i.e. it need not contain a transponder.

The optical components are object-oriented modeled, with classes that implement the functions of each element of the optical node. A graphic application of the model was developed in a way that allows a faithful reproduction of the functions of the optical elements and the visualization of system actions that follows.

The Figure 4a shows an optical node with the received lightpaths configuration (RX), from the control plane, and the operation status (TX). Other actions, such as fault simulations and configuration parameters of the amplifiers, transponders and OXC can be simulated. The figure 4b presents a simulated failure.

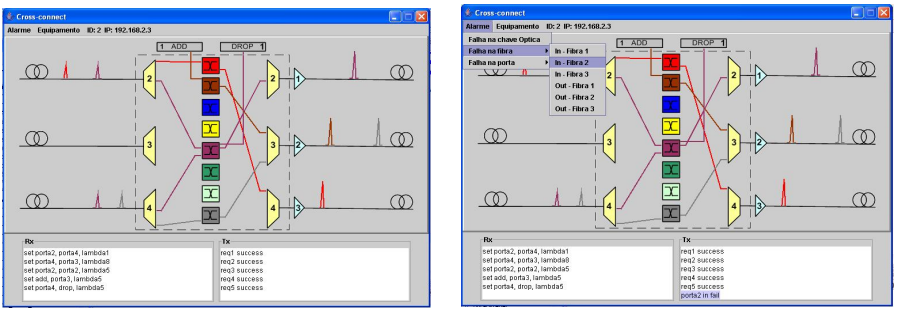


Fig. 4. a) Show screen of an optical node b) Fault simulated in the Testbed

Our controls system is composed of three basic protocols. One implements and creates optical pathways, the second manages the links, and the third activates an alternative route in case of fault (protection route). Figure 5 represents the basic structure of the physical site simulation.

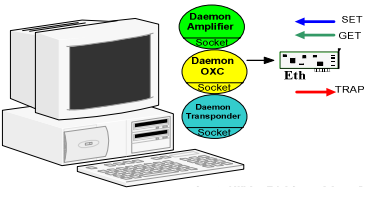


Fig. 5. The basic structure of the physical site simulation

Each physical element can simulate the following functions: on/off, optical ports read, alarm to power fault at in/out optical;

Faults are introduced in the system by using a graphical interface which activates the control system by sending a trap message. Specific actions are then launched by the control system.

2.5 Develop Environment

We used the Linux Fedora Release 2.0 distribution as our operational system,. The relational data base used was the PostgreSQL V8.1.2, and Apache Tomcat 4.1 was also used as our web server.

We used the environment KDE 3.5.2 with compiler GCC 3.4.6 and for the development of the functions of high performance. For Java programming it was used the eclipse-SDK-3.1.2-linux and J2SE 1.4.2_10 SDK environment.

To collect the network's information, monitoring of various package and filters, related with the protocol IP, it was used the Ethereal version 0.9.14 environment.

3 Testbed Working

3.1 Topology Discovery

The topology discovery is realized in a hybrid form. The management system sends a message Hello for its managed network nodes (the IP addresses of those nodes were obtained by using DHCP) so as to discover the active elements. The management system repeatedly sends hello messages each 30 (thirty) seconds to confirm the link stability of each element in the network.

In case three consecutive hello messages are not acknowledged by a node, a fault is assigned to it. This event is informed to the manager by a graphic interface.

Besides the detections of actives elements, the system has a configuration file, produced by the network's manager, which has the whole physical topology. When the activity of each optical element is detected, the system is assured that the network has no fault and that ligthpaths can be securely implemented.

3.2 Lightpaths Creation

To create a lightpath, a client, through the web interfaces, make a lightpath creation request to the management system. The management system analyzes the syntax of the solicitation and activates the control system, if the request is correct. Otherwise, it answers to the client informing a problem with the solicitation.

After a lightpath creating request, the control system uses the Routing and Wavelength Assignment - RWA algorithmi, based in Dijsktra, to create a lightpath using the First-Fit strategy to allocate a wavelength.

After it has the complete knowledge about the lightpath, the control system sends a message to each node, which belongs to the path, telling the node to switch its optical key with its respective wavelength. Then after the switching is proceeded the node responds to the control system. The control system consider the path as operational when it receives a right answer from all nodes in the path.

After that, the control system sends a message SETPORT to each node, which will answer with a SETPORTOK or a SETPORNOK message.

The message SETPORTOK sets a lightpath as active, to the control system. A SETPORTNOK answer makes the control system reject the realized request and make all keys to turn to the original states.

3.3 Lightpaths Deletion

For this operation, the management system asks the lightpath deletion to the control system, which, by its turn, identifies the requested path, its wavelength and respective nodes. So, the system control tells each optical cross connect to switch the optical keys for the original states by sending an UNSETPORT message.

After each switching, the node responds to the system control. The system control marks the lightpaths as deleted when all nodes respond with an UNSETPORTOK message.

The UNSETPORTOK message corresponds to turning the lightpath inactive and the UNSETPORTNOK message makes the control system disrespect the performed request and reset the keys to the original states.

3.4 Status Read

A read status contains the following information:

1. All network's nodes;
2. All IP addresses of the network's nodes;
3. The keys status of the OXCs;
4. All the activity lightpaths;
5. Basic information of each network device (OXCs, amplifiers and transponders). This information's are transmitted on XML format for web interface.

3.5 Errors Information's - Traps

Each activity element (OXC, amplifier or transponder) can perform fault simulations, by using traps. The following problems can be simulated:

1. on/off;
2. read errors on optical ports;
3. power fault on in/out optical ports;

4 Tests Results

We performed our tests with the help of a program which belongs to the management system. This program executes a random choice of the source and target nodes. Then, the system controls make the best choice for the lightpath (path and lambda) to be used.

On the tests, we made thirty random requests and the network saturated, on average, with twenty eight requests.

The Table 1 presents the results of the requests made to the management system. On the first column we have the number of request, on the second column the time in milliseconds to attend de request, on the third the number of nodes that belong to the lightpath and, in the last column, the complete lightpath (with node number, lambda, in/out optical port on each OXC).

Table 1. Lightpaths requests assisted by the system

1	235	2	5->4 - SERVICE@1* 5(3/2)+4(2/3)
2	319	2	3->4 - SERVICE@2* 3(3/1)+4(1/3)
3	304	3	3->4 - SERVICE@3* 3(3/0)+2(1/2)+4(0/3)
4	304	4	3->2 - SERVICE@1* 3(3/2)+5(1/0)+1(1/0)+2(0/3)
5	240	2	5->3 - SERVICE@2* 5(3/1)+3(2/3)
6	257	2	2->1 - SERVICE@2* 2(3/0)+1(0/3)
7	364	3	2->5 - SERVICE@1* 2(3/2)+4(0/2)+5(2/3)
8	247	2	4->3 - SERVICE@1* 4(3/1)+3(1/3)
9	304	3	1->3 - SERVICE@3* 1(3/1)+5(0/1)+3(2/3)
10	295	3	4->3 - SERVICE@4* 4(3/0)+2(2/1)+3(0/3)
11	351	3	3->5 - SERVICE@4* 3(3/1)+4(1/2)+5(2/3)
12	370	3	5->2 - SERVICE@3* 5(3/2)+4(2/0)+2(2/3)
13	275	2	1->5 - SERVICE@2* 1(3/1)+5(0/3)
14	286	3	3->2 - SERVICE@5* 3(3/1)+4(1/0)+2(2/3)
15	307	3	4->5 - SERVICE@3* 4(3/1)+3(1/2)+5(1/3)
16	297	3	5->3 - SERVICE@5* 5(3/2)+4(2/1)+3(1/3)
17	426	4	3->5 - SERVICE@6* 3(3/0)+2(1/0)+1(0/1)+5(0/3)
18	435	4	1->5 - SERVICE@5* 1(3/0)+2(0/2)+4(0/2)+5(2/3)
19	344	3	4->3 - SERVICE@6* 4(3/2)+5(2/1)+3(2/3)
20	448	4	5->4 - SERVICE@4* 5(3/0)+1(1/0)+2(0/2)+4(0/3)
21	269	4	5->4 - SERVICE@6* 5(3/0)+1(1/0)+2(0/2)+4(0/3)
22	260	4	1->5 - SERVICE@7* 1(3/0)+2(0/1)+3(0/2)+5(1/3)
23	194	3	1->4 - SERVICE@8* 1(3/1)+5(0/2)+4(2/3)
26	434	3	5->4 - SERVICE@7* 5(3/1)+3(2/1)+4(1/3)
28	306	4	3->1 - SERVICE@8* 3(3/1)+4(1/2)+5(2/0)+1(1/3)
30	476	2	4->2 - SERVICE@2* 4(3/0)+2(2/3)

In this specific test, with thirty random requests, twenty six were attended and four were denied due to network saturation. Then, we observe the need of an intelligent system for wavelength allocation. However, this problem is beyond the scope of this work.

The mean time for attending one request was 321 ms and the average number of hops was three.

The Table 2 presents the all packets used for attending the requests (twenty six). On this data we can view that the protocol consumed thirty percent of the whole traffic on the network.

Table 2. Total of packets captured at the test

Total packets captured on the network	207
Total UDP packets	78
Total UDP packets to node 1	11
Total UDP packets to node 2	13
Total UDP packets to node 3	16
Total UDP packets to node 4	19
Total UDP packets to node 5	19

The Figure 6 presents a graphic with the results of this test.

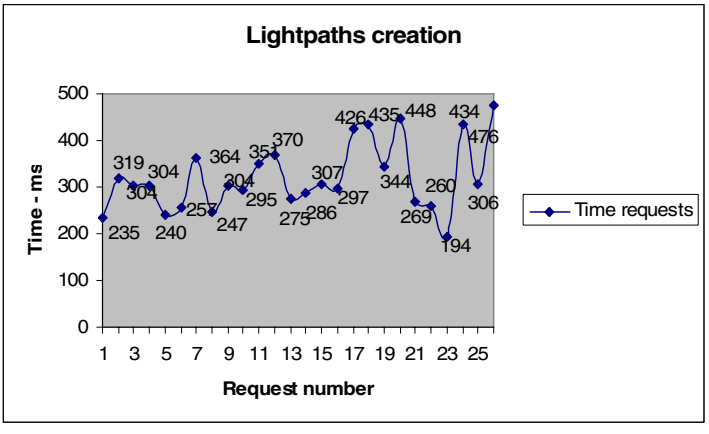


Fig. 6. Lightpaths requests results

5 Conclusions

This work presented the design and implementation of an optical transparent IP/WDM network testbed. Our testbed improves the security and malleability of the characterization of the transport, control and management planes of an optical network. We also developed a graphic user network interface for the client/management relation in our optical network. The transport plane model was implemented to represent transponders, amplifiers and OXCs. These are connected to the control and management planes through UDP/IP protocols using XML for supporting the information base. The control plane provides routing and protection in the event of data plane failures. The management plane has a functional structure based on fail management, on the network configuration program, on performance monitoring, on audit of changes and on the access security system. We obtained lightpaths creation, deletion and failure recovery in 352,79 ms, on average, in our testbed which has an IP based data link. Our testbed has high security control and

simple modeling using XML for the exchange of information. Because we used the UDP protocol, our solution possesses high performance and an intelligent provision of optical resources.

Acknowledgments. The authors wish to acknowledge the support of CNPq, CPqD and FINATEC/UnB-Brazil.

References

1. B. Rajagopalan, D. Pendarakis, R.S. Ramamoorthy, D. Saha, K. Bala, “IP over optical networks: Architectural aspects” IEEE Communications Magazine, no.9, pp. 94-102, (2000).
2. D. Cavendish, “Evolution of optical transport technologies: from SONET/SDH to WDM” IEEE Communications Magazine, no.6, pp.164-172, (2000).
3. R. E. Wagner et al., “MONET: Multiwavelength optical networking” IEEE JSAC, vol. 14, pp. 1349-1355, (1996).
4. Optical networking testbeds: experiences, challenges, and future directions (Part 1), Mohan, G.; Ghani, N.; Saradhi, C.V.; Stavdas, A. IEEE Optical Communications –Page(s): 52- 53 (2005)
5. W. Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Reading, MA: Addison-Wesley, (1996).
6. H.A.F. Crispim, Eduardo T. L. Pastor, H. Abdalla Jr, and A.J.M. Soares "Interface de Usuário para rede óptica IP/WDM em ambiente Web", XXII Simpósio Brasileiro de Telecomunicações - SBrT’– Brazil (2005).
7. Guangzhi Li, Jennifer Yates, D. Wang, Charles Kalmanek, “Control Plane Design for Reliable Optical Networks” IEEE Communications Magazine, pp. 90-96, (2002).

Voice Quality Management for IP Networks Based on Automatic Change Detection of Monitoring Data

Satoshi Imai, Akiko Yamada, Hitoshi Ueno,
Koji Nakamichi, and Akira Chugo

Fujitsu Laboratories Ltd, Kamikodanaka 4-1-1, Nakahara-ku, Kanagawa, Japan
{imai.satoshi, akikoo, ueno.hitoshi, nakamichi, chugo}@jp.fujitsu.com

Abstract. Recently, quality management of IP networks has become an important issue with the development of real-time applications such as IP-phones, TV-conferencing and video-streaming. Specifically, when voice data is mixed with various application data, there are worries that there will be a critical degradation in voice quality. In current management of voice quality, the general scheme is to monitor the voice quality using a preset threshold for the data. However, it's not easy to set a proper threshold for data monitoring in each case, because the characteristics of the monitoring data become more diverse as networks become larger in scale and more complex. We propose an automatic change detection scheme that detects changes in the behavior of the monitoring data with changes in the traffic conditions, without any preset threshold. The essential concept of our scheme is the application of a statistical scheme to sequential data monitoring. We also demonstrate the effectiveness of our proposed scheme when applied to voice quality management.

1 Introduction

With the sophistication of network technologies, the various application services are now deployed on IP networks. Specifically, the VoIP service is coming into general use as an alternative to the public switched telephone network (PSTN) call service. However, the emergence of these various IP services brings new issues to the VoIP service. The main issue is that the voice communication quality is extremely sensitive to the transmission characteristics of the voice packets. The VoIP quality is related to end-to-end delay, jitter and packet loss in the IP network, compared to other services such as E-mail and Web browsing.

- **End-to-end delay**
  is the time required for a voice packet sent by the caller to reach the callee. (This is equivalent to the difference between the packet arrival time at the callee and the packet timestamp that the caller puts on the transmission.) Real-time communication quality falls to a critical level when the delay exceeds a specific value (e.g. 150msec).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 451–460, 2006.
© Springer-Verlag Berlin Heidelberg 2006

- **Jitter**
  is the variation in packet arrival times. (This is equivalent to the difference between each end-to-end delay and the average end-to-end delay.)
  If the packet arrives too early or too late, the playout quality will be bad, because voice codecs require a steady packet stream to provide adequate playout quality.
- **Packet Loss**
  is a phenomenon in which voice packets are discarded on the IP network. Voice quality deteriorates because jumpiness and noise are caused by losing a part of the voice signals.

The behavior of these packet parameters depends on changes in the network conditions. In order to sustain VoIP quality, it is necessary to detect the change in network conditions before the VoIP quality deteriorates, and prevent any deterioration in quality.

Our study assumes an active measurement scheme by sending test packets as VoIP quality monitoring schemes, and focuses on a change detection scheme for network conditions (e.g. traffic load) based on analyzing the behavior of the monitoring data. In a change detection scheme, we should preset a suitable threshold to detect changes in the behavior of the monitoring data associated with a change in network conditions. However, as IP networks become large in scale and application traffic becomes more diverse, the behavior of the monitoring data (e.g. end-to-end delay and jitter) detected by active measurement will become nonidentical and more complex. Therefore, it is not easy to preset proper thresholds for data monitoring at all measurement points.

In this paper, we focus on the fact that jitter is extremely sensitive to changes in the network load, and propose a VoIP quality management scheme that detects the change in the behavior of the jitter associated with a change in the network load automatically and in real time by applying a statistical scheme.

The contents of this paper are organized as follows. Section 2 gives the VoIP quality management model and technical issues. Section 3 describes our scheme based on statistical tests. Section 4 presents some simulation results, and Section 5 concludes the paper with a summary and future studies.

2 VoIP Quality Management Model

Generally, the VoIP quality can be affected by even a sharp change in the traffic load which cannot be observed by MIB (which measures only the average load over several minutes duration). Therefore, we focus on an active measurement scheme which observes the behavior of the monitoring data associated with a sharp change in the traffic load, and define the VoIP quality management model shown in Fig. 1.

- The monitoring agents are located at each site, and observe the behavior of jitter on a peer-to-peer session between two agents.
- When an apparent change in jitter is detected, the monitoring agent advertises a change alarm to the management system or the network operators.

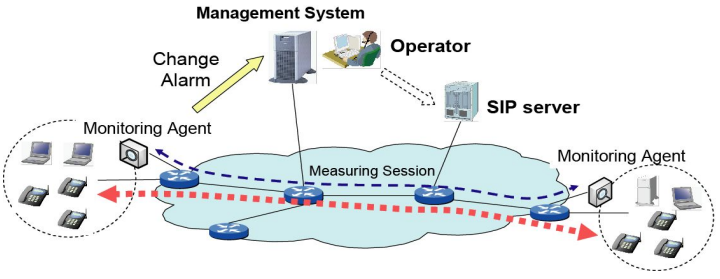


Fig. 1. VoIP Quality Management Model

– The management system or the network operators control the specific target (e.g. SIP server) based on the trigger of the change alarm.

2.1 Technical Issues

Recently, in the data mining fields, researches into change detection schemes for data monitoring abound [1,2]. It is a common issue to design proper thresholds for data monitoring in this fields. Representative schemes include statistics-based approaches, which are schemes that detect changes in the behavior of the monitoring data by i) defining the behaviors of the monitoring data in the normal states as a statistical model; ii) testing whether or not the present monitoring data belongs to a defined statistical model.

However, in order to analyze time-varying and complex data in real time, the data should be analyzed as a time series. Generally, for time-series data there is a strong correlation between past data and present data. Because statistics-based approaches require the assumption of statistical independence between sampled data, it is less effective to apply a direct statistics-based approach to time-series data.

Therefore, we propose a new automatic change detection scheme which has the following characteristics.

[1] Statistics-based change detection for time-series data
The proposed scheme focuses on the independent and normally-distributed residual error signals derived by modeling the time-series for jitter as an auto-regressive (AR) model [3]. As a result, we are able to use the statistics-based approach for time-series data by applying statistical tests to the residual errors in the AR model.

[2] Recursive calculation of statistical data
Every time a test packet is received, the proposed scheme calculates the jitter, and adaptively estimates the distribution of the residual errors in the AR model. Thus, we can detect changes in the behavior of the jitter using a statistical test in real time.

[3] Autonomic learning of normal behavior
The proposed scheme learns the normal behavior of the residual errors autonomously. As a result, the designers do not have to set any information about the monitoring data, such as the characteristics of the jitter, beforehand.

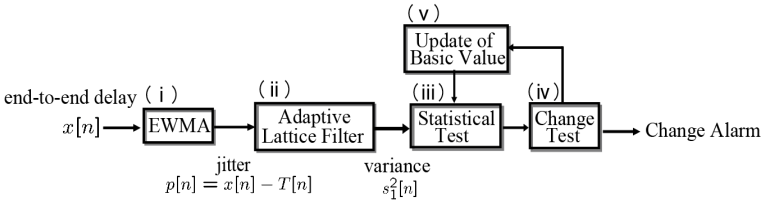


Fig. 2. Process Flow

3 Automatic Change Detection

In this section, we describe our proposed scheme. Our scheme can detect automatically and instantaneously changes in the behavior of the jitter measured by the monitoring agent. The algorithm is performed as follows. (Fig. 2)

- (**i**) Measure the end-to-end delay $x[n]$ in time n, and calculate the jitter $p[n]$ from $x[n]$ by an exponentially-weighted moving average (EWMA).
- (**ii**) Calculate the sample variance $s_1^2[n]$ of the residual errors in the AR model using an adaptive lattice filter [4,5].
- (**iii**) Calculate the statistical value and detect the instantaneous change (outlier) from the basic variance s_0^2 by the F-test [6].
- (**iv**) Detect any stationary change in $s_1^2[n]$.
- (**v**) Update the basic variance s_0^2 when a stationary change is detected.

The detailed algorithms on this flow are described below.

3.1 Time-Series Model for Jitter

We adopt the AR model as the time series model for the jitter $p[n]$, which is calculated from the end-to-end delay $x[n]$ measured in time $n \in \mathbf{Z}$. The AR model for the jitter p_t $(t = n - W + 1, \cdots, n)$ is defined by

$$p_t = -\sum_{i=1}^{m} a_i p_{t-i} + \epsilon_t. \tag{1}$$

where ϵ_t is the normally-distributed residual error with a mean of 0, and variance σ^2. W is the sampling data size for the estimation of AR parameters a_i $(i = 1, \cdots, m)$, the AR order m is a time-variable parameter which is optimally set by Akaike's Information Criterion (AIC) [3] in time n. In this paper, the jitter $p[n]$ is defined as $p[n] = x[n] - T[n]$, where the mean $T[n]$ of the end-to-end delay $x[n]$ is defined based on EWMA with the forgetting value $0 < \lambda << 1$, which gives

$$T[n] = (1 - \lambda)T[n-1] + \lambda x[n]. \tag{2}$$

Calculation of Residual Error. When the optimal AR model is estimated for the jitter, the residual errors become independent and normally-distributed white noise signals. In order to derive the optimal AR model, we use an adaptive lattice filter algorithm, which can calculate recursively for nonstationary data.

Adaptive Lattice Filter Algorithm

{Initial value};
for $0 \leq m < M$
$\gamma_m[-1] = \rho_m[-1] = \Delta_{m+1}[-1] = 0$
{Update of times};
for $0 \leq n$
$\gamma_0[n] = 0,\ \epsilon_0[n] = \rho_0[n] = p[n]$
$E_0^{\epsilon}[n] = E_0^{\rho}[n] = \omega E_0^{\epsilon}[n-1] + p[n]^2$
{Update of AR's orders};
for $0 \leq m < M$
$\Delta_{m+1}[n] = \omega \Delta_{m+1}[n-1] + \frac{\epsilon_m[n]\rho_m[n-1]}{1-\gamma_m[n-1]}$
$k_{m+1}^{\epsilon}[n] = \frac{\Delta_{m+1}[n]}{E_m^{\epsilon}[n]},\ k_{m+1}^{\rho}[n] = \frac{\Delta_{m+1}[n]}{E_m^{\rho}[n-1]}$
$\epsilon_{m+1}[n] = \epsilon_m[n] - k_{m+1}^{\rho}[n]\rho_m[n-1]$
$\rho_{m+1}[n] = \rho_m[n-1] - k_{m+1}^{\epsilon}[n]\epsilon_m[n]$
$E_{m+1}^{\epsilon}[n] = E_m^{\epsilon}[n] - k_{m+1}^{\rho}[n]\Delta_{m+1}[n]$
$E_{m+1}^{\rho}[n] = E_m^{\rho}[n-1] - k_{m+1}^{\epsilon}[n]\Delta_{m+1}[n]$
$\gamma_{m+1}[n] = \gamma_m[n] + \frac{\rho_m[n]^2}{E_m^{\rho}[n]}$
end.

The above-described parameter $(1-\omega) * E_m^{\epsilon}[n]$ with the forgetting value $0 << \omega < 1$ in the adaptive lattice filter algorithm is equivalent to the sample variance of residual error $\epsilon_m[n]$ in the number of samples defined by $W = 1/(1-\omega)$. In this paper, the order m to minimize AIC values expressed by

$$\mathbf{AIC}_m[n] = \min_m [\ \log((1-\omega) * E_m^{\epsilon}[n]) + \frac{2(m+1)}{1/(1-\omega)}\ ] \tag{3}$$

is defined as the optimal order $\hat{m}$. In our approach, we detect changes in the behavior of the jitter by observing this sample variance $(1-\omega) * E_{\hat{m}}^{\epsilon}[n]$.

3.2 Change Detection Scheme Based on a Statistical Test

The change detection scheme tests the change in the sample variance $(1-\omega) * E_{\hat{m}}^{\epsilon}[n]$ in the adaptive lattice filter algorithm, every time the end-to-end delay $x[n]$ is measured. To be more precise, the scheme automatically learns the basic sample variance s_0^2 and detects the differences between the basic sample variance s_0^2 and the present sample variance $s_1^2[n] = (1-\omega) * E_{\hat{m}}^{\epsilon}[n]$ by a statistical test.

Our statistical test scheme consists of testing two hypotheses: a null hypothesis (no change) and an alternative hypothesis (change), every time a test packet is received.

Outlier Detection. We treat the scheme which tests the statistical differences between two sample variances (s_0^2 and s_1^2) as a framework to detect any outlier to the sample variance (s_1^2) based on the basic sample variance (s_0^2).

The basic idea of outlier detection is described as follows. Based on the evidence that the residual errors in the optimal AR model belong to a normal distribution, we define basic n_0-samples of the residual errors as

$$[\textbf{Basic Samples}] \quad \epsilon_1^0, \cdots, \epsilon_{n_0}^0 \ \sim N\{0, \sigma_0^2\}, i.i.d \tag{4}$$

While, the present n_1-samples to test the change are defined as

$$[\textbf{Test Samples}] \quad \epsilon_1^1, \cdots, \epsilon_{n_1}^1 \ \sim N\{0, \sigma_1^2\}, i.i.d \tag{5}$$

Testing the differences between the sample variances s_0^2 from the basic samples and the sample variances s_1^2 from the test samples boils down to carrying out an F-test on the following hypotheses:

$$\begin{cases} \mathbf{H_0} : \sigma_0^2 = \sigma_1^2 \\ \mathbf{H_1} : \sigma_0^2 \neq \sigma_1^2 \end{cases} \tag{6}$$

The statistical value based on the null hypothesis ($\mathbf{H_0}$), which is expressed by

$$F_0 = \frac{\frac{n_1 s_1^2}{n_1 - 1}}{\frac{n_0 s_0^2}{n_0 - 1}} \tag{7}$$

belongs to an F-distribution with $(n_0 - 1, n_1 - 1)$ degrees of freedom as shown in Fig. 3. When $F^{\alpha}(n_0 - 1, n_1 - 1)$ is the upper α point on the F-distribution with $(n_0 - 1, n_1 - 1)$ degrees of freedom and $F_{\alpha}(n_0 - 1, n_1 - 1)$ is the lower α point, the F-test rule with significance level α are derived as below.

$$\begin{matrix} F_0 > F^{\alpha/2}(n_0 - 1, n_1 - 1) \\ F_0 < F_{\alpha/2}(n_0 - 1, n_1 - 1) \end{matrix} \Rightarrow \text{Accept the alternative hypothesis} \tag{8}$$

Because we premise on the basis that the both s_0^2 and s_1^2 are derived by the adaptive lattice filter, the sample variances s_1^2 are calculated in time n, and both n_0 and n_1 are equivalent to $1/(1-\omega)$. Therefore, the outlier detection rules based on the basic variance s_0^2 with significance level α are defined as

$$\frac{s_1^2[n]}{s_0^2} > F^{\alpha/2}\left(\frac{\omega}{1-\omega}, \frac{\omega}{1-\omega}\right) \Rightarrow \text{An upper outlier has occurred} \tag{9}$$

$$\frac{s_1^2[n]}{s_0^2} < F_{\alpha/2}\left(\frac{\omega}{1-\omega}, \frac{\omega}{1-\omega}\right) \Rightarrow \text{A lower outlier has occurred} \tag{10}$$

Updating the Basic Variances. The update rule for the basic variance s_0^2 is described as below. We expand a "scheme to detect an outlier" to a "scheme to detect a stationary change". The update rule is

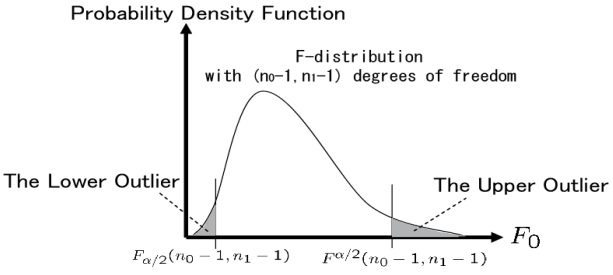


Fig. 3. F-distribution

– Update basic variance s_0^2 when a stationary change is detected.

When k_1 and k_2 are the numbers of upper and lower outlier detections respectively, and N is the trial number of the outlier detection test, the rate of upper and lower outlier detection is shown below.

$$r_1 = k_1/N, \quad r_2 = k_2/N \tag{11}$$

We assume that

– if the rate of outlier detection r_1 or r_2 is under the significance level $\alpha/2$, the basic variance s_0^2 is correct.
– if the rate of outlier detection r_1 or r_2 is much over the significance level $\alpha/2$, the basic variance s_0^2 is changed.

The above assumption are represented as the following hypotheses.

$$\text{Upper Side Change} : \begin{cases} \hat{\mathbf{H}}_\mathbf{0} : r_1 \leq \frac{\alpha}{2} \\ \hat{\mathbf{H}}_\mathbf{1} : r_1 > \frac{\alpha}{2} \end{cases} \tag{12}$$

$$\text{Lower Side Change} : \begin{cases} \hat{\mathbf{H}}_\mathbf{0} : r_2 \leq \frac{\alpha}{2} \\ \hat{\mathbf{H}}_\mathbf{1} : r_2 > \frac{\alpha}{2} \end{cases} \tag{13}$$

The problem of testing these hypotheses can be resolved using statistical values which are expressed by

$$Z_1 = \frac{r_1 - \alpha/2}{\sqrt{\alpha/2(1-\alpha/2)/N}}, \; Z_2 = \frac{r_2 - \alpha/2}{\sqrt{\alpha/2(1-\alpha/2)/N}} \tag{14}$$

If the trial number N is sufficiently large, the statistical values (Z_1 and Z_2) belong to a standard normal distribution. When the upper $\tilde{\alpha}$ point of a standard normal distribution is $Z^{\tilde{\alpha}}$, the rules to detect whether or not a stationary change has occurred are as expressed below.

$$Z_1 > Z^{\tilde{\alpha}} \Rightarrow \text{ stationary change has occurred on the upper side} \tag{15}$$

$$Z_2 > Z^{\tilde{\alpha}} \Rightarrow \text{ stationary change has occurred on the lower side} \tag{16}$$

When a stationary change is detected by the above-described rules, we update the basic variance s_0^2 to

- $s_0^2 * F^{\alpha/2}\left(\frac{\omega}{1-\omega}, \frac{\omega}{1-\omega}\right) \Leftarrow$ change on the upper side
- $s_0^2 * F_{\alpha/2}\left(\frac{\omega}{1-\omega}, \frac{\omega}{1-\omega}\right) \Leftarrow$ change on the lower side

In our approach, the estimation of the both rates in Equation(11) are determined in real time. If $D[n]$ is a discrete parameter that is 1 when an outlier is detected, and 0 when no outlier is detected, the rate $r[n]$ in time n is calculated using the forgetting value $0 < \eta \ll 1$ as below.

$$r[n] = (1 - \eta)r[n - 1] + \eta D[n] \tag{17}$$

Where the trial number N is equivalent to about $1/\eta$. In this paper, we set η as $\alpha/2$ based on the significance level for outlier detection.

4 Simulation Analysis of Voice Quality Monitoring

In this section, we evaluate the performance of the proposed scheme on a simulation model that is designed by OPNET. In the simulation, two scenarios are implemented: one scenario is the case with only VoIP traffic, and with a G.711 codec on the network, the other is the case with mixed traffic which includes VoIP, FTP and HTTP traffic on the network. From the simulation results, we show that the proposed scheme enables us to detect sharp changes in the traffic load indirectly by analyzing the behavior of the jitter as below.

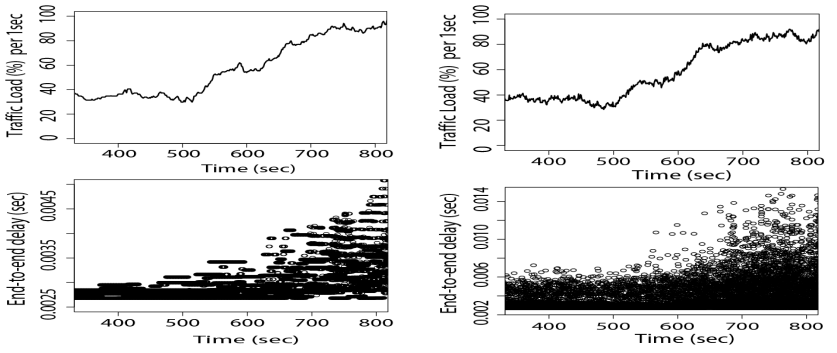
4.1 Simulation Scenarios

When the traffic load per 1 [sec] on the measurement route is increased after 500 [sec] from the stationary load of about 30% in both scenarios, the time-series graphs of the end-to-end delay measured by the monitoring agent are as shown in Fig. 4. The proposed scheme is applied to the end-to-end delay in Fig. 4(a) and Fig. 4(b), respectively. In this paper, we use the forgetting values of $\lambda = 0.01$, $\omega = 0.99$, $\eta = 5e - 7$, and the significance levels in the statistical test of $\alpha = 1e - 6$, $\tilde{\alpha} = 0.01$.

Relationship Between Load and Jitter. The jitter, which is calculated from the end-to-end delay, has the following characteristics as shown in Fig. 5.

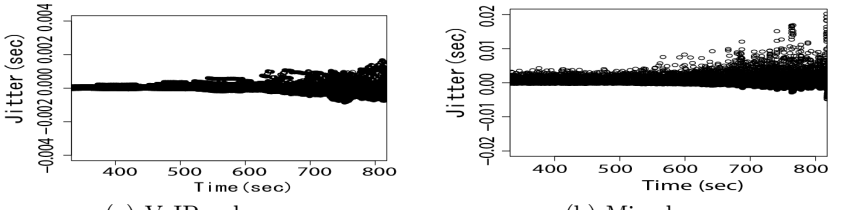
- As the traffic load increases, the jitter also increases.
- The jitter in the mixed traffic case is larger than the jitter in the case with only VoIP traffic.

Because our scheme detects behavioral changes in the jitter associated with sharp changes in the traffic load by observing the behavior of the residual errors in the AR model, we show the relationship between the traffic load and the residual errors. The relationship between the means of the traffic load and the variances of the residual errors per 1 [sec] are shown in Fig. 6. The correlation coefficients below an about 90% load in Fig. 6(a) and Fig. 6(b) are 0.912 and 0.872, respectively. These results show a strong correlation between the traffic load and the variances of the residual errors in both scenarios. Therefore, it is effective to observe the variances of the residual errors in order to detect sharp changes in the traffic load.



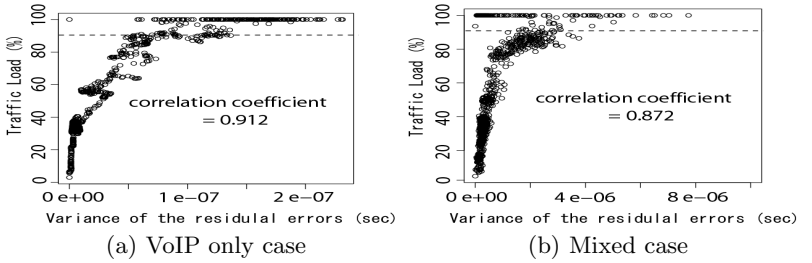
(a) VoIP only case (b) Mixed case

Fig. 4. Time-series graphs (Upper: traffic load, Lower: end-to-end delay)



(a) VoIP only case (b) Mixed case

Fig. 5. Jitter calculated from the end-to-end delay in Fig.4

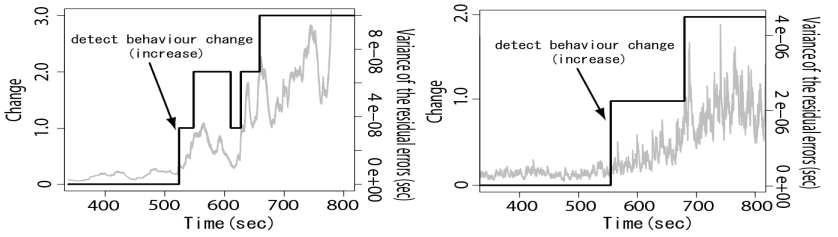


(a) VoIP only case (b) Mixed case

Fig. 6. Relationships between the means of the traffic load and the variance of the residual error per 1 [sec]

Detection Results. The detection results for the end-to-end delay in Fig. 4 are shown in Fig. 7. The change values in these results are the sum of the change amounts, which indicate +1 when it is detected that the jitter increased, and −1 when it is detected that the jitter decreased.

In the simulation results, we can verify that our scheme detects that the variances of the residual errors (i.e. the behavior of the jitter) have changed after the traffic load changes sharply, for example, at about 520 [sec] in Fig. 7(a) or at about 550 [sec] in Fig. 7(b). Therefore, our scheme enables us to detect automatically the changes in the behavior of the jitter associated with the changes in



(a) VoIP only case (b) Mixed case

Fig. 7. Simmulation results of automatic change detection

the traffic load, without presetting a threshold for the jitter, based on recursive calculations.

5 Conclusion

In this paper, we focus on the measurement of jitter by a monitoring agent, and propose a change detection scheme based on time-series modeling and a statistical approach, which detects automatically and instantaneously changes in the behavior of jitter associated with sharp changes in the traffic load. From the simulation results, we were able to verify that our scheme is very effective as a scalable system, because it can be operated without presetting any information about the monitoring data.

Future studies will focus on research and development of a control scheme to avoid any deterioration in VoIP quality based on this change detection scheme.

References

1. Lui, C.L., Fu, T.C, and Cheung, T.Y.: Agent-Based Network Intrusion Detection System Using Data Mining Approaches. Third International Conference on Information Technology and Applications (ICITA), Vol. 1. (2005) 131-136
2. Petrovsky, M. I.: Outlier Detection Algorithms in Data Mining Systems, Programming and Computer Software, Vol. 29, Issue. 4. (2003) 228-237
3. Chatfield, C.: The Analysis of Time Series: An Introduction, Sixth Edition. Chapman & Hall/CRC. (2004)
4. Friedlander, B.: Lattice Filters for Adaptive Processing. Proceedings of IEEE, Vol. 70. (1982) 829–867.
5. Singer, A.C.: Universal Linear Prediction by Model Order Weighting. IEEE Trans, Signal Process, Vol. 24, No. 10. (1999) 2685–2699.
6. Bulmer, M.G.: Principles of Statistics. Dover Publications, Inc. (1979)

Parameter Design for Diffusion-Type Autonomous Decentralized Flow Control

Chisa Takano[1,2], Keita Sugiyama[2], and Masaki Aida[2]

[1] Traffic Engineering Division, NTT Advanced Technology Corporation
[2] Graduate School of System Design, Tokyo Metropolitan University

Abstract. We have previously proposed a diffusion-type flow control mechanism as a solution for severely time-sensitive flow control required for high-speed networks. In this mechanism, each node in a network manages its local traffic flow on the basis of only the local information directly available to it, by using predetermined rules. In addition, the implementation of decision-making at each node can lead to optimal performance for the whole network. Our previous studies show that our flow control mechanism with certain parameter settings works well in high-speed networks. However, to apply this mechanism to actual networks, it is necessary to clarify how to design a parameter in our control mechanism. In this paper, we investigate the range of the parameter and derive its optimal value enabling the diffusion-type flow control to work effectively.

1 Introduction

The rapid spread of the Internet will necessitate the construction of higher-speed backbone networks in the near future. In a high-speed network, it is impossible to implement time-sensitive control based on collecting global information about the whole network because the state of a node varies rapidly in accordance with its processing speed although the propagation delay is constant. If we allow sufficient time to collect network-wide information, the data so gathered is too old to use for time-sensitive control. In this sense, each node in a high-speed network is isolated from up-to-date information about the state of other nodes or that of the overall network.

This paper focuses on a flow control mechanism for high-speed networks. From the above considerations, the technique used for our flow control method should satisfy the following requirements: (i) it must be possible to collect the information required for the control method, and (ii) the control should take effect immediately.

There are many other papers reporting studies on flow control optimization in a framework of solving linear programs [1,2,3]. These studies assume the collection of global information about the network, but it is impossible to achieve such a centralized control mechanism in high-speed networks. In addition, solving these optimization problems requires enough time to be available for calculation,

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 461–470, 2006.
© Springer-Verlag Berlin Heidelberg 2006

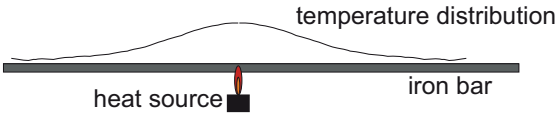


Fig. 1. Example of thermal diffusion phenomena

so it is difficult to apply these methods to decision-making on a very short time-scale. Therefore, in a high-speed network, the principles adopted for time-sensitive control are inevitably those of autonomous decentralized systems.

Decentralized flow control by end hosts, including TCP, is widely used in current networks, and there has been a lot of research in this area [3,4]. However, since end-to-end or end-to-node control cannot be applied to decision-making on a time-scale shorter than the round-trip delay, it is inadequate for application to support decision-making on a very short time-scale. In low-speed networks, a control delay on the order of the round-trip time (RTT) has a negligible effect on the network performance. However, in high-speed networks, the control delay greatly affects the network performance. This is because the RTT becomes large relative to the unit of time determined by node's processing speed, although the RTT is itself unchanged. This means that nodes in high-speed networks experience a larger RTT, and this causes an increase in the sensitivity to control delay. To achieve rapid control on a shorter time scale than the RTT, it is preferable to apply control by the nodes rather than by the end hosts.

We therefore considered a control mechanism in which the nodes in a network handle their local traffic flows themselves, based only on the local information directly available to them. This mechanism can immediately detect a change in the network state around the node and apply quick decision-making. Although decision-making at a local node should lead to action suitable for the local performance of the networks, the action is not guaranteed to be appropriate for the overall network-wide performance. Therefore, the implementation of decision-making at each node cannot lead to optimum performance for the whole network.

In our previous studies, we proposed diffusion-type flow control (DFC) [5,6,7]. DFC provides a framework in which the implementation of the decision-making of each node leads to high performance for the whole network. The principle of our flow control model can be explained through the following analogy [7]. When we heat a point on a cold iron bar, the temperature distribution follows a normal distribution and heat spreads through the whole bar by diffusion (Fig. 1). In this process, the action in a minute segment of the iron bar is very simple: heat flows from the hotter side towards cooler side. The rate of heat flow is proportional to the temperature gradient. There is no communication between two distant segments of the iron bar. Although each segment acts autonomously, based on its local information, the temperature distribution of the whole iron bar exhibits orderly behavior. In DFC, each node controls its local packet flow, which is proportional to the difference between the number of packets in the

node and that in an adjacent node. Thus, the distribution of the total number of packets in a node in the network becomes uniform over time. In this control mechanism, the state of the whole network is controlled indirectly through the autonomous action of each node.

Our previous studies show that our flow control mechanism with certain parameter settings works well in high-speed networks. However, to apply DFC to actual networks, it is necessary to clarify how to design parameters in our control mechanism. This is one of central issues to be solved for applying DFC to actual networks. In this paper, we investigate the appropriate value of a parameter in DFC and propose a design policy of the value.

2 Preliminary

2.1 Diffusion-Type Flow Control Mechanism

In the case of Internet-based networks, to guarantee end-to-end quality of service (QoS) of a flow, the QoS-sensitive flow has a static route (*e.g.*, RSVP). Thus, we assume that a target flow has a static route. In addition, we assume all routers in the network can employ per-flow queueing for all the target flows.

In DFC, each node controls its local packet flow autonomously. Figure 2 shows the interactions between nodes (routers) in our flow control method, using a network model with a simple 1-dimensional configuration. All nodes have two incoming and two outgoing links, for a one-way packet stream and for feedback information, that is, node i ($i = 1, 2, \dots$) transfers packets to node $i + 1$, and node $i + 1$ sends feedback information to node i. For simplicity, we assume that packets have a fixed length in bits.

When node i receives feedback information from downstream node $i + 1$, it determines the transmission rate for packets to the downstream node $i + 1$ using the received feedback information, and it adjusts its transmission rate towards the downstream node $i + 1$. The framework for node behavior and flow control is summarized as follows:

- Each node i autonomously determines the transmission rate J_i on the basis of only the local information directly available to it, that is, the feedback information obtained from the downstream node $i + 1$ and node i's information.
- The rule for determining the transmission rate is the same for all nodes.
- Each node i adjusts its transmission rate towards the downstream node $i + 1$ to J_i. (If there are no packets in node i, the packet transmission rate is 0.)
- Each node i autonomously creates feedback information according to a predefined rule and sends it to the upstream node $i - 1$. Feedback information is created periodically with a fixed interval τ_i.
- The rule for creating the feedback information is the same for all nodes.
- Packets and feedback information both experience the same propagation delay.

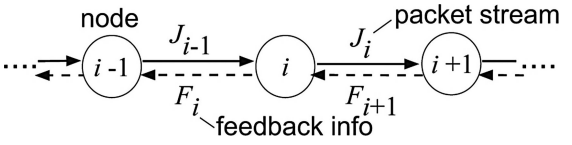


Fig. 2. Node interactions in our flow control model

As mentioned above, the framework of our flow control model involves both autonomous decision-making by each node and interaction between adjacent nodes. There is no centralized control mechanism in the network.

Next, we explain the details of DFC. The transmission rate $J_i(\alpha, t)$ of node i at time t is determined by

$$J_i(\alpha, t) = \max(0, \min(L_i(t), \tilde{J}_i(\alpha, t))), \quad \text{and} \tag{1}$$

$$\tilde{J}_i(\alpha, t) = \alpha\, r_i(t - d_i) - D_i\,(n_{i+1}(t - d_i) - n_i(t)), \tag{2}$$

where $L_i(t)$ denotes the value of the available bandwidth of the link from node i to node $i + 1$ for target flow at time t, $n_i(t)$ denotes the number of packets in node i at time t, $r_i(t - d_i)$ is the target transmission rate specified by the downstream node $i + 1$ as feedback information, and d_i denotes the propagation delay between nodes i and $i + 1$. Determination of $L_i(t)$ is explained later. In addition, $r_i(t - d_i)$ and $n_{i+1}(t - d_i)$ are reported from the downstream node $i + 1$ as feedback information with propagation delay d_i. Parameter α (≥ 1), which is a constant, is the flow intensity multiplier. Parameter D_i is chosen to be inversely proportional to the propagation delay [6] as $D_i = D/d_i$, where D (> 0), which is a positive constant, is the diffusion coefficient.

The feedback information $\mathbf{F}_i(t)$ created every fixed period τ_i by node i consists of the following two quantities:

$$\mathbf{F}_i(t) = (r_{i-1}(t), n_i(t)). \tag{3}$$

Node i reports this to the upstream node $i - 1$ with a period of $\tau_i = d_{i-1}$. Here, the target transmission rate is determined as $r_{i-1}(t) = J_i(1, t)$. Moreover, the packet flow $J_i(t)$ in node i is renewed whenever feedback information arrives from the downstream node $i + 1$ (with a period of $\tau_{i+1} = d_i$).

To enable an intuitive understanding, we briefly explain the physical meaning of DFC. We replace i with x and apply a continuous approximation. Then the propagation delay becomes $d_i \to 0$ for all i and the flow (2) is expressed as

$$\tilde{J}(\alpha, x, t) = \alpha\, r(x, t) - D\,\frac{\partial n(x, t)}{\partial x}, \tag{4}$$

and the temporal evolution of the packet density $n(x, t)$ may be represented by a diffusion-type equation,

$$\frac{\partial n(x, t)}{\partial t} = -\alpha\,\frac{\partial r(x, t)}{\partial x} + D\,\frac{\partial^2 n(x, t)}{\partial x^2}, \tag{5}$$

using the continuous equation $\partial n(x,t)/\partial t = -\partial \tilde{J}(\alpha,x,t)/\partial x$. As explained in Sec. 1, our method aims to perform flow control using the analogy of diffusion. We can expect excess packets in a congested node to be distributed over the whole network and expect normal network conditions to be restored after some time.

In addition to the above framework, we consider the boundary condition of the rule for determining the transmission rate in the DFC. It is explained in [7].

Moreover, we should appropriately adjust available bandwidth $L_i(t)$ among flows, since the link bandwidth is shared by multiple flows. The value of $L_i(t)$ is determined by dividing the link bandwidth proportionally to $\tilde{J}_i(\alpha,t)$ among flows [8].

3 Parameter Design

3.1 Approach

In DFC, there are two important parameters: one is the flow intensity multiplier α and the other is the diffusion coefficient D. Our previous study shows that $\alpha = 1$ is a natural and appropriate choice because that means the balance between input and output traffic at a node. The residual problem is to determine an appropriate value of D. The diffusion coefficient governs the speed of diffusion. In physical diffusion phenomenon, larger D causes faster diffusion. If DFC model is completely corresponding to physical diffusion phenomenon, a large value of D is suitable for fast recovery from congestion. Unfortunately, DFC is not completely corresponding to physical diffusion. As we see later in the next section, too large value of D in DFC blocks diffusion phenomenon in networks. The reason of this problem comes from the fact that networks have discrete configurations although physical diffusion phenomenon occur in a continuous space-time environment. That is, the spatial configuration of routers is discrete, and timing of control actions is also discrete.

Conversely, too small value of D causes very slow diffusion, and this means that stolid congestion recovery wastes much time.

Our approach to design a value of D is simple. We take a larger value of D in the range of values in which diffusion can occur in networks.

3.2 Range of Diffusion Coefficient and Parameter Design

The partial differential equation (5) describes temporal evolution of packet density in continuous approximation of networks. The first term on the right-hand side in (5) describes a stationary packet flow and this is not concerned with diffusion, but the second term is essential in diffusion. Thus, we consider the following partial differential equation,

$$\frac{\partial n(x,t)}{\partial t} = D\,\frac{\partial^2 n(x,t)}{\partial x^2}, \tag{6}$$

where this is the ordinary diffusion equation.

Of course, the structure of networks and the timing of control actions are not continuous. Behaviour of DFC is described by a difference equation rather than the differential equation. In other words, DFC make networks solve a difference equation with discrete space x and discrete time t.

For simplicity, we assume all the links in networks have same length Δx. In this situation, interval of DFC's action is the same for all node, and we denote it as Δt. The difference equation corresponding to (6) is as follows:

$$\frac{n(x, t+\Delta t) - n(x,t)}{\Delta t} = D\,\frac{n(x+\Delta x,\, t) - 2\,n(x,\, t) + n(x-\Delta x,\, t)}{(\Delta x)^2}. \tag{7}$$

If the solution of (7) exhibit similar behavior to that of (6), DFC appropriately works and diffusion of packet density occurs. Our issue is to find appropriate value of D in which the solution of (7) exhibits diffusion phenomenon.

Let node position in 1-dimensional configuration be x_k ($x_{k+1} - x_k = \Delta x$; $k = 0, 1, \ldots, S$), and time of DFC's action be t_ℓ ($t_{\ell+1} - t_\ell = \Delta t$; $\ell = 0, 1, \ldots, T$). We take the boundary condition, $n(t,\, x_0) = n(t,\, x_S) = 0$. If behavior of $n(x_k,\, t_\ell)$ exhibits a diffusion effect with time,

$$\lim_{\ell \to \infty} n(x_k,\, t_\ell) = 0, \tag{8}$$

for all k. In general, $n(x_k,\, t_\ell)$ satisfying the boundary condition is represented as the following Fourier series,

$$n(x_k,\, t_\ell) = \sum_{m=0}^{\infty} n_m(x_k,\, t_\ell), \quad \text{and} \quad n_m(x_k,\, t_\ell) = A_{m,\ell}\,\sin\left(\frac{k\,m\,\pi}{S}\right), \tag{9}$$

where $A_{m,\ell}$ is a time-dependent coefficient. If (8) is valid in any cases,

$$\lim_{\ell \to \infty} n_m(x_k,\, t_\ell) = 0 \tag{10}$$

for all non-negative integers m. By substituting (9) into (7), we have

$$A_{m,\ell} = A_{m,\ell-1}\,\left(1 - \frac{4\,D\,\Delta t}{(\Delta x)^2}\sin^2\frac{m\,\pi}{2\,S}\right) = \cdots = A_{m,0}\,\left(1 - \frac{4\,D\,\Delta t}{(\Delta x)^2}\sin^2\frac{m\,\pi}{2\,S}\right)^{\ell}.$$

Therefore,

$$n_m(x_k,\, t_\ell) = A_{m,0}\,\left(1 - \frac{4\,D\,\Delta t}{(\Delta x)^2}\sin^2\frac{m\,\pi}{2\,S}\right)^{\ell}\sin\left(\frac{k\,m\,\pi}{S}\right). \tag{11}$$

From (10), D should satisfy

$$\left|1 - \frac{4\,D\,\Delta t}{(\Delta x)^2}\sin^2\frac{m\,\pi}{2\,S}\right| < 1 \tag{12}$$

and we obtain the range of the diffusion coefficient D,

$$0 < D < \frac{1}{2}\frac{(\Delta x)^2}{\Delta t}. \tag{13}$$

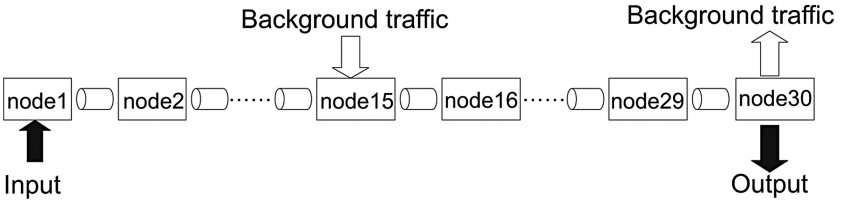


Fig. 3. Simulation model

We set the length of links as $\Delta x = 1$ and the interval of DFC's control action (it is equal to the propagation delay of a link) as $\Delta t = 1$, the range of the diffusion coefficient D is

$$0 < D < \frac{1}{2}. \tag{14}$$

Consequently, to make fast diffusion, we take a value of D as large as possible in this range.

4 Simulation Results

In this section, we show simulation studies about the performance of DFC with different values of the diffusion coefficient D in order to verify the range (14) and our design policy of D. Simulations were made by using ns2 simulator [9]. We extended the simulation tool ns2 capability with the function of DFC.

4.1 Simulation Model

Figure 3 shows our network model with 30 nodes, which is used in the simulations. Although this 1-dimensional model looks simple, it represents a part of a network and describes a path of the target end-to-end flow extracted from the whole network. Propagation delay of each link between nodes is 0.1 ms, and the capacity of buffer at each node is 1800 packets. For simplicity, the lengths of all links are the same (generalization to inhomogeneous link lengths is possible [6]). A packet has a fixed length of 1500 Bytes and the link bandwidth is $1,000,000$ packets/s. This means the link bandwidth is 12 Gbps. Note that only the bandwidth-delay product for a link is an essential parameter in this situation. If we choose the propagation delay of 0.01 ms, the link bandwidth is 120 Gbps.

If we represent the lengths of links by their delays, and the length of links is 1.0 unit time, then $\Delta x = \Delta t = 1$, and the range of D is (14). Hereafter, D is represented in this unit system.

The simulation scenario was as follows. There were two TCP flows. The target flow is between node 1 and node 30, while the background traffic flows between node 15 and node 30. The maximum TCP window size of both flows is 5,000 and it is chosen as sufficiently larger than the bandwidth-delay product of RTT. The target flow and the background flow start at simulation time $t = 0$ s and

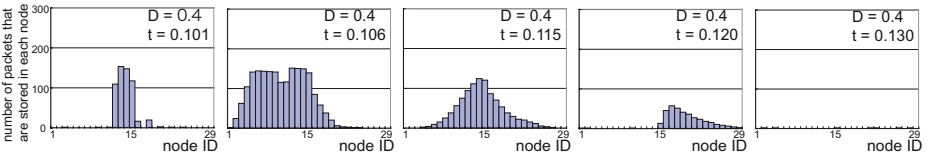


Fig. 4. Temporal evolution of distribution of packets stored in each node ($D = 0.4$)

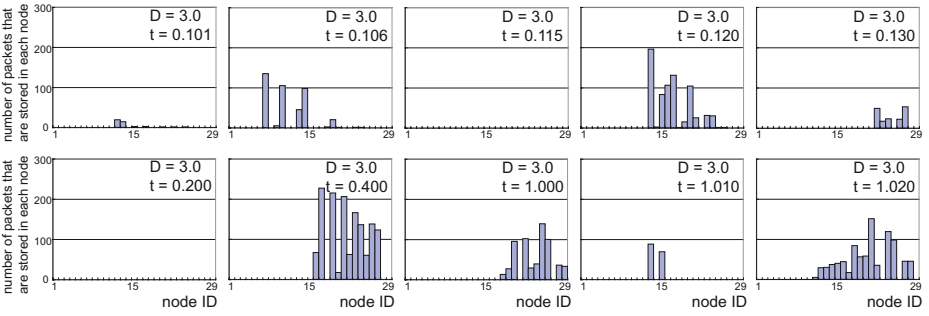


Fig. 5. Temporal evolution of distribution of packets stored in each node ($D = 3.0$)

$t = 0.1$ s, respectively. After the background flow traffic entered the network, the link from node 15 to 16 became a bottleneck, and traffic of both flows was regulated by predefined rules for DFC. After congestion occurred, we investigated the temporal evolution of the network state.

4.2 Simulation Results

To clarify the diffusion effect by DFC, we show temporal evolution of the number of stored packet in each node.

Figures 4 and 5 show the results obtained from our simulatiopns with $D = 0.4$ and 3.0, respectively. The horizontal axes denote node ID (1–29) and the vertical axes denote the number of stored packet at the node. Here, we omit node 30 since it has no stored packets. Simulation time is shown in each graph.

From Fig. 4, after the time when the background traffic enters (after 0.1 s), the network prevents the stored packets centralizing at a certain node. This effect is from DFC. The distribution of the number of packets exhibits orderly behavior. In this case, the diffusion coefficient $D = 0.4$ is in the range of (14).

Incidentally, if DFC was not applied, all stored packets are at node 15 [8]. This causes packet losses and the reduction of TCP window size. By introducing DFC, each node cooperatively acts to avoid packet losses even though decision-making of each node is based only on the local information.

On the other hand, Fig. 5 does not show orderly behavior of the packet distribution. That is, DFC with $D = 3.0$ does not exhibit diffusion effect. In addition, it does not recover from the congestion even though DFC with $D = 0.4$ recovers quickly from congestion. In this case, the diffusion coefficient $D = 3.0$ is out of the range of (14).

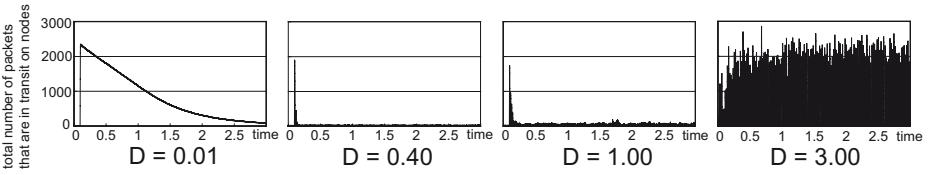


Fig. 6. Temporal evolution of the total number of packets that are stored in nodes

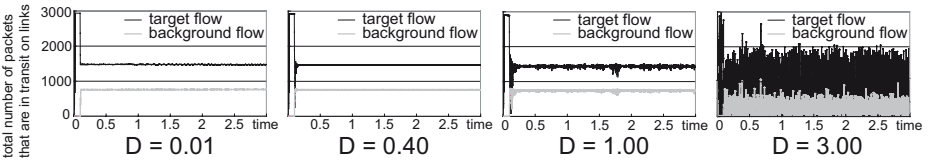


Fig. 7. Temporal evolution of the total number of packets that are in transit on links

Figure 6 denotes the temporal evolution of the total number of packets that are stored in nodes, in cases of $D = 0.01$, 0.4, 1.0 and 3.0, respectively. The horizontal axes denote simulation time and the vertical axes denote the total number of packets that are stored in nodes. The first two cases are in the range of (14). Both cases exhibit diffusion effect and the total number of packets decreases with time, but it is very slow in the case of $D = 0.01$. This means that too small D prevents fast recovery from congestion.

The last two cases in Fig. 6 are out of the range of (14). Larger value of $D > 1/2$ causes instability of the total number of packets. In particular, too large D exhibits chaotic behavior.

Next, we investigate how much packets are transmitted in the network. The volume of packet transmission at time t can be denoted by the total number of packets in transit on links of the network. Figure 7 shows the results in cases of $D = 0.01$, 0.4, 1.0 and 3.0. The horizontal axes denote simulation time and the vertical axes denote the total number of packets that are in transit on links.

Since the maximum number of packets in transit on a link at a moment was 100, and the background flow passed through about half of links of the target flow, the maximum total number of target flow's packets in transit on links was $2,900$ when $t \leq 0.1$ and was $1,450$ when $t > 0.1$. On the other hand, the maximum total number of background flow's packets in transit on links was 750 after $t = 0.1$. From the first two panels in Fig. 7, the numbers of packets in transit on links for both flows reached almost their maximums in a short time and these results mean that they fairly share the link bandwidth. For larger D that is out of the range of (14), the total number of packets in transit on links becomes unstable. These results show that larger value of $D > 1/2$ degrades the performance of packet transmission.

These simulation results substantiate our design policy of D; to make fast diffusion, we take a value of D as large as possible in the range of (14).

5 Conclusions

To overcome the difficulty in control of high-speed networks, we have proposed DFC. In this control mechanism, the state of the whole network is controlled indirectly through the autonomous action of each node; each node manages its local traffic flow on the basis of only the local information directly available to it, by using predetermined rules. By applying DFC, the distribution of the total number of packets in each node in the network becomes uniform over time, and it exhibits orderly behavior. This property is suitable for fast recovery from congestion.

One of important issues in design of DFC is how to choose the value of diffusion parameter. This is the central issue for enabling DFC to make diffusion effects of packet density in networks. This paper investigates the appropriate value of the diffusion parameter.

We determined the range of the diffusion parameter by applying the condition for discrete space-time computations of the diffusion equation to DFC. On the other hand, even if the value is in the range, too small value of the diffusion parameter causes very slow diffusion, and this means that stolid congestion recovery makes to waste much time. Consequently, to make fast diffusion, we should take a value of the diffusion parameter as large as possible in this range. Simulation results verified our proposed design policy.

This research was partially supported by the Grant-in-Aid for Scientific Research (S) No. 18100001 (2006–2010) from the Japan Society for the Promotion of Science.

References

1. Y. Bartal, J. Byers, and D. Raz, "Global optimization using local information with applications to flow control," Proc. the 38th Ann. IEEE Symp. on Foundations of Computer Science, pp.303–312, Oct. 1997.
2. S. H. Low and D. E. Lapsley, "Optimization flow control-I: basic algorithm and convergence," IEEE/ACM Trans. Netw., vol.7, no.6, pp.861–874, 1999.
3. J. Mo and J. Walrand, "Fair end-to-end window based congestion control," IEEE/ACM Trans. Netw., vol.8, no.5, pp.556–567, Oct. 1999.
4. R. Johari and D. Tan, "End-to-end congestion control for the Internet: Delays and stability," IEEE/ACM Trans. Netw., vol.9, no.6, pp.818–832, Dec. 2001.
5. C. Takano and M. Aida, "Stability and adaptability of autonomous decentralized flow control in high-speed networks," IEICE Trans. Commun., vol.E86-B, no.10, pp.2882–2890, 2003.
6. C. Takano, M. Aida, and S. Kuribayashi, "Autonomous decentralized flow control in high-speed networks with inhomogeneous configurations," IEICE Trans. Commun., vol.E87-B, no.6, pp.1551–1560, 2004.
7. C. Takano and M. Aida, "Diffusion-type autonomous decentralized flow control for end-to-end flow in high-speed networks," IEICE Trans. Commun., vol.E88-B, no.4, pp.1559-1567, 2005.
8. M. Aida, C. Takano and A. Miura, "Diffusion-type flow control scheme for multiple flows," The 19th International Teletraffic Congress, pp. 133-142, 2005.
9. The Network Simulator—ns-2.
`http://www.isi.edu/nsnam/ns/`

Bandwidth Management for Smooth Playback of Video Streaming Services

Hoon Lee[1], Yoon Kee Kim[2], and Kwang-Hui Lee[1]

[1] Changwon National University, Changwon, Korea
{hoony, khlee}@changwon.ac.kr
[2] KT R&D Center, Seoul, Korea
yoonkee@kt.co.kr

Abstract. In this work we propose an analytic framework for managing the optimal bandwidth required for a smooth playback of a stored video streaming service. First, we argue that the mean buffering delay in the node is minimized if the video packets are paced to a fixed interval at the beginning of the transfer. Second, we argue that an appropriate level of bandwidth has to be maintained so that neither buffer-underflow nor buffer-overflow occurs, via which a smooth playback of video stream can be maintained. From the numerical evaluation, we found that the paced packet stream experiences smaller delay than the original random packet stream. Next, we computed a range of bandwidth for a video stream so that neither buffer-underflow nor buffer-overflow occurs, and we showed that paced packet stream requires smaller bandwidth than the original random packet stream.

1 Introduction

Recently, it has become usual that one can view a stored video file such as news clips from high-speed Internet [5]. The bandwidth of a news clip in [5] with screen size of 6.5 by 5 inches is 492Kbps, from which we could witness frequent freezing of scenes even for this narrow bandwidth video file due to limited bandwidth allocated to a flow.

As the bandwidth of the down link for the end users in the current access network increases from a few megabits per second (Mbps) to tens or hundreds of Mbps, we envision that a need for a high-quality VoD (Video on demand) or IPTV (Internet protocol TV) service increases in the near future.

If the bandwidth in an end-to-end path of a video stream is sufficient, the video file will be played back smoothly after the initial buffering. However, when the network is congested or if the bandwidth of a video server or user is not sufficiently provided, it is inevitable that a degradation of the quality of service (QoS) will occur by repeated freezing of the scenes due to frequent buffering. To resolve this problem, we have to devise a method to manage bandwidth for a video stream so that a smooth playback of a video file is provided to the users.

It is known that the delay of a sequence of packets transferred through the IP network has a great spectrum of variations. The unpredictable behavior of packets

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 471 – 480, 2006.
© Springer-Verlag Berlin Heidelberg 2006

inside the network causes a serious quality degradation problem to a receiving side of a streaming service such as VoD or music. The random packet delay results in random arrivals of packets at the receiving side, which causes buffer underflow or overflow. Frequent packet underflow causes frequent buffering at the video decoder, and consequently playback is not carried out smoothly. One can also note that some packets are lost inside the network as well as at the receiving end when the buffer overflows. Packet loss requires a retransmission of the lost packets from the source or concealment of loss at the receiver, which disrupts smooth decoding of a video stream.

Summarizing those two causes of QoS degradation for video streaming services in IP network, we argue that a method to avoid the major sources of QoS degradation in a packet-based video streaming has to be devised.

One can avoid this problem if one adopts a sophisticated service scheme such as resource reservation at the access side as well as the differentiated packet scheduling scheme architecture at the core of a network for a service with strict QoS requirements. However, those service architectures require modification in the network equipments at the network access and core, and so it is envisioned that those service architectures will not be widely deployed in the near future.

One can think of CAC (connection admission control), adaptive video source coding, dynamic bandwidth allocation or dimensioning method to guarantee QoS to the users. There exist a lot of research works on the former two schemes [13]. Under the current simple service structure of best effort IP network, it would not be efficient to control the video source rate frequently based on the microscopic dynamism of bandwidth consumption for each flow. Therefore, network operators may want a simple network operation that relies not on the complex control of the associated packets but on a network provisioning with an appropriate amount of bandwidth.

To that purpose, we will devise a bandwidth dimensioning and management method to a video streaming service, because the loss and delay problem in packet-based network that operates in best effort scheme can be resolved by provisioning at least an appropriate bandwidth for a flow [11].

We argue that the delay and its variation of a packet in the network can be minimized if we devise a traffic control scheme to the packet stream with random inter-arrival time such that the packet arrival pattern of a video stream behaves almost in packet train manner[1].

In order to illustrate our argument in the context of IP network, we have two purposes in this work. First, we propose a traffic control model called the packet pacing scheme for IP access network and compare the delay performance between the original random packet stream and the paced packet stream via approximate queuing model. Second, we apply the model into the bandwidth dimensioning for the streamed video delivery service network.

The contribution of the work is as follows: First, we propose a packet pacing scheme in the video streaming server so that the video stream becomes smooth and the mean delay becomes minimized. Via numerical experiment, we show that the delay performance of paced packet stream is better than that of original random packet stream by numerical experiment. Second, we propose a bandwidth management

[1] Packet train means that consecutive flows of packets are almost equally distanced.

scheme for a video stream service by managing the buffer level so that neither buffer starvation nor buffer overflow occurs. Via numerical experiment, we illustrate the efficacy of the proposed method in the design of the optimal link capacity of video server.

This paper is organized as follows: In Section 2 we describe the concept of pacing the packet stream. In Section 3 we present a queuing model that represents the expected queuing delays for the two packet streams, the randomly spaced packet stream and the paced packet stream. In Section 4 we apply the proposed model to managing the bandwidth for video streaming server. In Section 5 we present the result of a numerical experiment and discuss the implication of our argument. Finally, in Section 6, we summarize the paper.

2 Pacing the Packet in a Video Streaming Service

First, let us define a video streaming service, which refers to a transfer of a live or stored video content via a streaming technology in the network. A data stream is played back as it is being delivered to the receiver. This is also called a progressive download in that users watch the video while download is going on [3]. To be more specific, streaming refers to a transfer of data via circuit-switched transfer technologies such as asynchronous transfer mode network, whereas progressive download refers to a transfer of data via packet-switched transfer technology such as IP (Internet protocol). We will assume an IP network as a basis of our discussion.

An architecture of the video streaming service is briefly depicted in Fig.1, where a video server is connected to an IP backbone via a packet pacer. The role of packet pacer is to store the arriving packets from the video server into a queue, and it transfers them to the access router that is located at the entrance of IP backbone in a constant time interval, say d. A discussion on the optimal value of d will be given later.

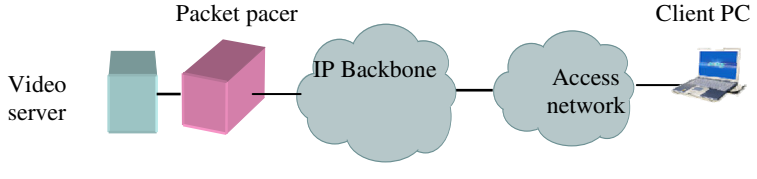


Fig. 1. Architecture of video streaming services

On the other hand, IP backbone network is assumed to be equipped with a transparent data transfer protocol such as an MPLS (Multi-protocol Label switching) tunnel with fine grain QoS and CAC, via which packets from the packet pacer is transmitted to the access network of a client in a secure and seamless manner in terms of security as well as the QoS. It is usual that the current IP backbone network is over-provisioned, so we consider that there exists almost no buffering delay to a stream of packets that flows inside the IP backbone network especially when packets are served by an MPLS tunnel between the edge-to-edge routers of IP backbone network. One can find an example of the deployment of this concept from [15].

Finally, the access network of client in the receiving side may be composed of DSL, E-PON (Ethernet passive optical network), or HFC (hybrid fiber and coaxial). Fortunately, the traffic stream that flows down to the receiver from the access network rarely meets collision due to the inherent broadcasting mechanism for the down traffic. Therefore, the concern for the QoS of streaming video lies in the sending side of the IP backbone network, specifically the video server side. As such, an appropriate level of sustainable throughput has to be provided at the access network of a video server in order that users can watch a video over the IP network with a satisfactory QoS.

3 Delay Analysis

Under the prevalent environment of the current Internet with a slim pipe at the access and a fat pipe at the core of the network, it can be easily found that no advantages are obtained by having non-identical shapers for a flow at each node in an end-to-end path of a flow in order to make smooth the traffic flow. Thus, it is recommended that shaping is carried out only once at the ingress point of the network [6]. If we follow this argument, the end-to-end delay of a packet is composed of the shaping delay and the sum of nodal delays at the traversing nodes as well as the propagation delay and some hardware processing delays at the sender and receiver. A detailed discussion about the component of end-to-end delay is out of the scope of this work, and refer to [10] for more information about it. We focus only on the buffering delay of the packet pacer in order to investigate the efficacy of pacing to the expected waiting time of a packet for a video streaming service.

Let us assume that the original packet stream is generated by Poisson distribution with mean arrival rate λ and variance of inter-arrival time σ_A^2. The service time of the packet is generally distributed with mean service time $1/\mu$ and variance σ_S^2. The mean offered load to the system is given by $\rho=\lambda/\mu$. Let us define the squared coefficient of variation for the inter-arrival time C_A^2 and the service time C_S^2 respectively, are given by $C_A^2 = \lambda^2\sigma_A^2$ and $C_S^2 = \mu^2\sigma_S^2$. Then, the original packet stream in the buffer of the packet pacer is modeled as an M/G/1 queuing system with FIFO (First in first out) packet services. Note that the mean delay performance of M/G/1 queuing system is well-known [4], [8]. Using the result from [4] for the mean waiting time W_O of an arriving packet from the original packet stream, we obtain (1).

$$W_O = \frac{1}{\lambda}\left(\frac{\rho^2}{1-\rho}\times\frac{1+C_S^2}{2}\right) \tag{1}$$

Now let us discuss about the effect of pacing the original packet stream into an equi-distant packet stream with inter-packet arrival time defined by d. The original packet stream with Poisson arrival process is transformed to a general independent arrival process with constant packet inter-arrival time d, where $d = 1/\lambda$, so that the effective arrival rate of the video source is the same for both models. Then, the paced arrival process is D/G/1 queue with squared coefficient of variation for the inter-arrival time C_A^2 to be zero.

Note that D/G/1 queue is a special case of GI/G/1 queuing system with fixed packet arrival time. For the GI/G/1 queuing system, an approximation formula for the mean waiting time of an arriving packet is given in [4]. If we modify the result of [4] by changing the general independent arrival process to a constant inter-arrival time process, we obtain the following result for the mean waiting time W_P of the paced packet.

$$W_P \approx \frac{\rho/\mu}{1-\rho} \times \frac{C_S^2}{2} \times \exp(-\frac{2}{3} \cdot \frac{1-\rho}{\rho} \frac{1}{C_S^2}) \tag{2}$$

If we compare the two formulas for W_O and W_P, the mean waiting time of M/G/1 queue and D/G/1 queue, respectively, we can find that the following inequality holds if and only if $\rho<1$.

$$W_P < W_O \tag{3}$$

Therefore, we can argue that the paced packet stream experiences smaller delay than the original random packet stream. This is the first desired aspect of our proposition.

4 Sustainable Load and Optimal Bandwidth for a Video Stream

One of the hot issues in video streaming service via Internet is the guarantee of sustained bandwidth[2] to a video session so that consecutive frames are transferred in a stable manner between video source and sink over the public IP network [2]. If the above requirement is not met, the quality of video will be degraded due to frame loss or delayed frame arrival. One more problem that has to be taken into account is the problem of delay and jitter under the dynamic network load condition. When frames arrive to the receiver in very random intervals the receiver has to prepare a playback buffer in order to absorb the variance of delays. If frames arrive in predictable and stable manner, the size of playback buffer will be reduced. The problem of delay jitter at the receiving side can be alleviated by buffering some amount of packets at the playback buffer at the receiver. The problem of packet starvation can be resolved by providing a sufficient bandwidth to a video flow so that packets sent from the source are arrived to the receiver without severe delay inside the network.

If we investigate the basic principle of video delivery in commercial IP network, we can find that almost all the video streaming services are based on download-before-viewing scheme. In this case, the video server partitions the video frame into packets, stores them into a buffer, and sends them through the network, while the receiver plays back the frames packed from the received packets while packets from the later frames are still being downloaded, so that playback and delivery of frames are carried out simultaneously. If too small bandwidth is allocated to the video server, the frames will be transmitted to the receiver later than the required frame rate, and the video quality will be degraded due to starvation. On the other hand, if too large bandwidth is provisioned to a flow at the server, early-arrived frames have to wait at

[2] Sustained bandwidth means the amount of bandwidth that can be provided continuously to a flow by a network during the session duration time.

the playback buffer in the receiver. Therefore, management of bandwidth to an appropriate level is very important in video streaming delivery network.

Let us remind that a video server is connected to a commercial IP backbone network via a packet pacer, where the interval of packet arrival time is kept to be constant. At a client PC a playback buffer exists and packets transferred from the video server via the commercial IP network are stored and played back at a proper rate. In order to playback the video stream at a proper rate, sufficient amount of packets have to be stored a priori at the playback buffer. Therefore, one has to make sure that some amount of packets is stored at the buffer of a client PC so that the video frames can be successfully reconstructed [9]. Note that we can regard the buffer level at the server and receiver in the same manner if we assume that the IP backbone is transparent to the packets.

Let $Q(t)$ be the number of packets that are stored in a buffer at time t. Let K be the capacity of the buffer. Note that the size of K should not necessarily to be large; it must be large enough to store a few packets that constitute a video frame. Note that buffer overflow occurs when $Q(t)$ exceeds the buffer capacity K, in which case the packets that have arrived during the overflow period will be lost. Note also that buffer underflow occurs when $Q(t)=0$, during which no frame construction can be carried out.

On the other hand, one may note that neither buffer-overflow nor buffer-underflow must occur if the video frames are to be reconstructed completely in real-time. This necessitates a management scheme in the buffer in such a way that the buffer occupancy is kept to a certain level so that neither buffer-overflow nor buffer-underflow occurs.

The buffer control scheme is aimed as follows: Let us assume two thresholds, the upper limit U and the lower limit L of the buffer occupancy. We aim that the offered load, which is a function of the inter-packet arrival time d, is controlled in such way that the buffer occupancy in steady state, which is denoted by Q, is kept between the upper limit U and the lower limit L, which is represented by $L \leq Q \leq U$.

Using the Little's formula for the mean length of a queue for the paced packet stream, we can obtain (4).

$$L \leq \frac{\rho^2 \times C_S^2}{2(1-\rho)} \exp\left(-\frac{2(1-\rho)}{3\rho \times C_S^2}\right) \leq U \tag{4}$$

Note that we can tune the offered load and the arrival rate of the traffic if the values U and L are determined. On the other hand, the arrival rate of the traffic can be controlled by tuning inter-packet gap d that we have defined in the previous section.

We can find that the formula (4) is a non-linear inequality, and we can obtain the solution by using a numerical method. However, one can not see an explicit relation between the system parameters and the acquired performance if one uses the numerical method. Therefore, let us resort to an analytic approximation method so that we can obtain a closed formula for the optimal sustainable load of the system, via which we can obtain an intuition for the optimal load level of a system from assuming a small number of parameters that we have defined up to now. From the preliminary numerical experiment we found that the solution obtained by analytic approximation method is closely matched with that obtained by numerical method.

Let us define a variable X as $X = \frac{1-\rho}{\rho}$, and let us also define χ as $\chi = \frac{C_S^2}{2}$. Then we can rewrite (4) into (5) by using Maclaurin series and some manipulation, which is given as follows.

$$L \le \frac{\chi - \frac{1}{3}X}{X(1+X)} \le U \tag{5}$$

Note that the formula (5) is composed of two inequalities: the left-hand inequality (LHI) and the right-hand inequality (RHI), which are given as follows.

$$\begin{aligned} LHI &: LX^2 + (L + \frac{1}{3})X - \chi \le 0, \\ RHI &: UX^2 + (U + \frac{1}{3})X - \chi \ge 0 \end{aligned} \tag{6}$$

We can obtain a closed form solution for the inequalities (6), which is given as follows.

$$\begin{aligned} X_L &\le \frac{\sqrt{(L+1/3)^2 + 4L\chi} - (L+1/3)}{2L}, \\ X_R &\ge \frac{\sqrt{(U+1/3)^2 + 4U\chi} - (U+1/3)}{2U} \end{aligned} \tag{7}$$

Note that we can also obtain a range of an optimal load for the video delivery service system from (7). Eq. (8) represents two loads, the upper bound and lower bound of the load, which is represented by ρ_U^P and ρ_L^P, respectively. In the sequel, ρ_X^Y stands for the load of a video stream, and the lower index $X=L$ stands for the lower bound and $X=U$ stands for upper bound, and the upper index $Y=O$ stands for the original stream and $Y=P$ stands for the paced stream.

$$\begin{aligned} \rho_U^P &= \frac{1}{1+X_R}, \\ \rho_L^P &= \frac{1}{1+X_L} \end{aligned} \tag{8}$$

Note from (7) and (8) that the upper and low bounds for the load of video traffic are functions of the buffer threshold as well as the squared coefficient of variation for the service time of the packet. The former parameter is controlled by a network operator, whereas the latter one is generic to the traffic source. Note that we can estimate the range of the required bandwidth for a video flow from (8).

As we have argued before, the two most effective ways to sustain a satisfactory level of the QoS for the video services is as follows: First, shape the arrival pattern of the video packets to be constant by packet pacing. Second, keep the range of the offered load of the video stream between ρ_L^P and ρ_U^P. In this respect, the above result gives us a very useful insight for the provisioning of a video streaming service.

5 Numerical Results

First, let us compare the range of the offered load for the original video stream and paced video stream, and verify the validity of our argument in (3). Note that we have presented the load for the paced video stream in (8). The upper and lower loads, ρ_U^O and ρ_L^O, respectively, for the original video stream can be also obtained in the same analogy that we have used in deriving the result (8), which is shown in (9).

$$\rho_L^O = \frac{\sqrt{L^2 + 4\eta L} - L}{2\eta}, \qquad \rho_U^O = \frac{\sqrt{U^2 + 4\eta U} - U}{2\eta} \tag{9}$$

where η is defined by $\eta = \frac{1 + C_S^2}{2}$.

In order to compare the performance between the original video stream and the paced video stream, we carry out the experiment by assuming the same threshold parameters for the buffer as well as the traffic sources. Let us assume that the average number of frames generated by a video source is 15frames per second. A video frame is represented by 250×200pixel, and a pixel is represented by one byte. Then, the mean data rate of a video source is 6Mbps, which is denoted by $r = 6$Mbps.

As to the packet traffic model, let us assume that a video frame is composed of a number of packets with Gaussian distributed packet length. We use the traffic source parameters that have been discussed in [7]. The mean packet length is assumed to be 50bytes and maximum packet size of 125bytes, and the standard deviation of the packet length is 25bytes. The lower limit L of the buffer is assumed to be one packet, so that $L=1$, while the upper limit U of the buffer is assumed to be a design parameter, which varies from 2 to 47.

Fig.2 illustrates the sustainable load of the system, which is an upper bound of load. One can find that the upper limit on the offered load of a paced video stream is higher than that of original video stream, which means that the delay performance of paced stream is better than that of original stream, which complies with our first argument.

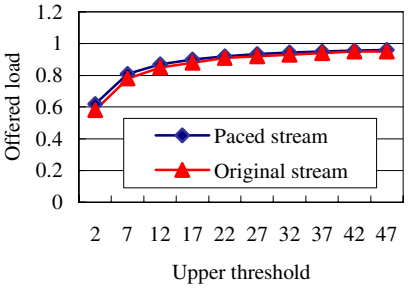


Fig. 2. Upper bound of load

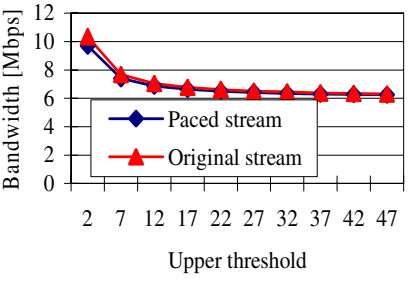


Fig. 3. Required bandwidth

From the numerical experiment, we found that the lower bound of the offered load is ρ_L^P =0.5 for the paced stream and ρ_L^O =0.46 for the original stream, which correspond to the minimum load that has to be sustained in the system so that buffer starvation does not occur. This means that the offered load of a playback buffer has to be kept no smaller than 0.5 for the paced stream and 0.46 for the original stream, otherwise the buffer suffers starvation. From the above result, we can also estimate the required bandwidth C^P for a paced video stream as well as C^O the original video stream under the assumed system parameters. The result is given in Fig.3.

We found that, in order to avoid the buffer starvation and overflow, the bandwidth of a video server has to be maintained between C_L^P =6.25Mbps and C_U^P =12Mbps, respectively, for a video flow with packet pacing, whereas the bandwidth of a video flow without packet pacing has to be maintained between C_L^O =6.30Mbps and C_U^O =13.04Mbps. Here, the lower index L and U stands for lower and upper bound, respectively. From this result we obtain the following conclusions: *The smoother the traffic stream, the smaller the required bandwidth*. This is the final finding of this work.

From Fig.3 one can also find that the required bandwidth of a video stream decreases as the upper threshold of a buffer increases. Therefore, we can conclude that there exists a trade-off between the required bandwidth and the buffer capacity for a video stream service, and one has to prepare a sufficient buffer space, say about a few tens of packets, if one wants to avoid over-provisioning of bandwidth to a video streaming service.

Note that one has to focus more on the lower limit of the bandwidth, because the lower limit of the bandwidth is the minimum bandwidth that has to be sustained to a video stream so that the video frames are played back in a smooth manner without starvation in the playback buffer. On the other hand, the upper limit of the bandwidth can be used as a reference for determining the size of the buffer.

6 Conclusions

In this work we proposed a packet transfer control and bandwidth management scheme for a smooth playback of a stored video stream over IP network. To that purpose, we first argued that the packet pacing guarantees a minimum delay to packets, whereas the bandwidth management scheme guarantees smooth playback of video frames. By using queuing model and numerical experiment, we validated our argument. Next, we proposed a buffer management scheme for guaranteeing smooth playback of streamed video data, and we obtained an explicit formula for the optimal level of offered load for a video streaming service. Finally, we presented a quantitative result for the manageable range of bandwidth for sustained quality of video streaming service by assuming typical parameters for the traffic source.

The intuition obtained from this work is that a better quality for the video stream can be provided by sustaining an optimal level of bandwidth to a video flow. We also showed that the delay of packets from the video source can be reduced a priori by pacing the packets from the video server when the backbone network is assumed to be transparent of delay to the transit packets. The result can be also applied to the

allocation of bandwidth for an IP TV service in which the compressed video frames are transferred in a CBR-like manner via Internet.

In the future we will investigate the validity of the proposed method via simulation or real-field experiment. We will also elaborate on the sophistication of the video source model, via which we can obtain a more practical intuition for the operation and management of the network resources for the video streaming services in the future Internet as well as the current best effort Internet.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments, which improved the presentation of the paper.

References

1. Acharya S., Smith B., Parns P., Characterizing user access to video on the World Wide Web, Proc. ACM/SPIE MMCN Jan. 2000.
2. Apostolopoulos J.G., Video communications and video streaming, Streaming media systems group, Hewlett-Packard Laboratories, May 1, 2000.
3. Besset C., Le Drogo C., Dumetz C., Paquette R., Orange video project with Alcatel, Alcatel Telecommunications Review 4th Quarter, 2005.
4. Bolch G., Greiner S., de Meer H., Trivedi K., Queueing Networks and Markov Chains, John Wiley & Sons, Inc. 1998.
5. http://www.chosun.com/tv/news/
6. Georgiadis L., Guerin R., Peris V., Sivarajan K., Efficient network QoS provisioning based on per node traffic shaping, IEEE/ACM Transactions on Netwoking, Vol.4, No.4, August 1996.
7. Kim D.-H., Jun K., Dynamic bandwidth allocation scheme for video streaming in wireless cellular networks, IEICE Trans. Commun., Vol. E89-B, No.2, February 2006.
8. Kleinrock L., Queuing systems, Volume 1: Theory, John Wiley & Sons, 1975.
9. Komori Y., Kasahara S., Sugimoto K., A study on dynamic rate control mechanism based on end-user level QoS for streaming services, Technical Report of IEICE NS 2003-332 (2004-03).
10. Lee H., Back Y.-C., Anatomy of delay for voice services in NGN, Proceedings of Fall Conference of the Communication Society of IEEK, 2003, Korea.
11. Lee H., Sohraby K., Flow-aware link dimensioning for guaranteed-QoS services in broadband convergence networks, Paper submitted to JCN, March 2006.
12. Sivaraman V., Chiussi F., Gerla M., Traffic shaping for end-to-end delay guarantees with EDF scheduling, Proceedings of IWQoS 2000.
13. Wu D.P., Hou Y.W., Zhang Y.Q., Scalable video coding and transport over broadband wireless networks, Proc. IEEE, 2001, 89.
14. Le Boudec J.-Y., Network calculus made easy, Technical report EPFL-DI 96/218, Dec. 1996.
15. Sardella A., Video transit on an MPLS backbone, A solution brief from Juniper networks, Juniper Networks, Inc., 200106-0

An Enhanced RED-Based Scheme for Differentiated Loss Guarantees*

Jahwan Koo[1], Vladimir V. Shakhov[2], and Hyunseung Choo[1,**]

[1] School of Information and Communication Engineering, Sungkyunkwan University
Chunchun-dong 300, Jangan-gu, Suwon 440-746, South Korea
jhkoo@songgang.skku.ac.kr, choo@ece.skku.ac.kr
[2] Institute of Computational Mathematics and Mathematical Geophysics of SB RAS
Prospect Akademika Lavrentjeva 6, Novosibirsk 630090, Russia
shakhov@skku.edu

Abstract. Recently, researchers have explored to provide a queue management scheme with differentiated loss guarantees for the future Internet. The Bounded Random Drop (BRD), known as the best existing scheme, is one of such efforts which guarantees strong loss differentiation to the level of traffic in the different service classes. Even though BRD has several benefits such as low complexities and good functionalities, we identify that it has some shortcomings such as low throughput, long queuing delays, and selection problem of optimal values of parameters. Specifically, the shortcomings stem from calculating drop probabilities based on the arrival rate estimation and dropping incoming packets randomly with calculated drop probabilities without considering the current buffer occupancy. A multiple queue management scheme based on differential drop probability, called MQDDP, is proposed to overcome BRD's shortcomings as well as support absolute loss differentiation. This scheme extends the original Random Early Detection (RED), recommended by IETF for next generation Internet gateways, into multiple class-based queues by deriving the drop probability equations based on a queueing model. We also compare MQDDP to BRD for high traffic intensity and show that MQDDP has the better performance in terms of packet drop rate.

1 Introduction

Today's Internet has grown from a small data transfer-oriented network to a large public accessed multi-service network. Various types of real time and non-real time traffic with varying requirements are transmitted over the Internet. With the growth of the Internet, it has become necessary to deploy appropriate

* This research was supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005-(C1090-0501-0019).

** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 481–490, 2006.
© Springer-Verlag Berlin Heidelberg 2006

solutions to provide different levels of service in terms of throughput, delay, jitter, and loss guarantees. As a result, there have been a number of previous works for providing some form of service differentiation. The proportional differentiated services (PDS) model [3] is one of such efforts.

Most mechanisms for PDS have been used proprietary algorithms for delay and loss differentiations. For example, proportional delay differentiation has been implemented with appropriate scheduling algorithms [4] [8] [2] and proportional loss differentiation has been implemented by queue management algorithms [5] [1]. Many of the existing PDS mechanisms pursues relative QoS requirements rather than absolute guarantees. This often results in significant variations in the actual level of performance, in particular, across periods of high and low loads. Recently, some mechanisms for providing both absolute loss and delay guarantees and proportional differentiations were reported in [7] and [6]. However, they have been conflicted with the requirements for the scalability in the Internet due to their additional complexities.

Specifically, we limit our discussion to differentiated and absolute loss guarantees. More recently, the Bounded Random Drop (BRD) [6], known as the best existing scheme among the relevant ones, has been reported to offer service differentiations while keeping complexity low. Even though BRD has several benefits such as low complexities and good functionalities, we identify that it has some shortcomings such as low throughput, long queuing delays, and selection problem of optimal values of parameters. This paper focuses on improving the throughput since BRD shows low throughput which seems a shortcoming of the method. The proposed scheme is based on the properties of the original Random Early Detection (RED) recommended by IETF for next generation Internet gateways with a mechanism for absolute loss guarantees.

The rest of the paper is organized as follows. Section 2 briefly reviews the previous works. In section 3, we describe the shortcoming of BRD and propose a multiple queue management scheme based on differential drop probability, MQDDP, that support differentiated loss guarantees for multiple service classes. In section 4, we compare MQDDP to BRD for high traffic intensity and show that MQDDP has the better performance in terms of packet drop rate. The final section offers some concluding remarks.

2 Related Work

We briefly review the previous works in this section. As mentioned earlier, many related works have been proposed to meet the demand for differentiated loss guarantees in the future Internet. Specifically, the proportional loss differentiation (PLD) model is one of such efforts, which offers fixed proportions on loss rates between the QoS levels of the different classes rather than absolute bounds. Dovrolis *et al.* [4] who developed first the PLD model, claim that several previous mechanisms such as complete buffer partitioning (CBP), partial buffer sharing (PBS), or multi-class RED, are not suitable for relative differentiated services. They propose and evaluate Proportional Loss Rate (PLR) mechanisms

that closely approximate the PLD model. Liebeherr *et al.* [7] propose a novel algorithm called Joint Buffer Management and Scheduling (JoBS) for the integration of buffer management and packet scheduling and compare with PLRs for PLD service. Li *et al.* [8] propose a novel algorithm by employing Probabilistic Longest Queue First (PLQ) mechanism and claim that its implementation cost is much less than PLRs and even more practical. Zeng *et al.* [10] propose more enhanced dropping algorithm than PLRs in terms of the packet shortage phenomenon.

More recently, the Bounded Random Drop (BRD) proposed in [6] offers service differentiation while keeping the complexity low. The performance goals of its scheme are to achieve both absolute and relative loss requirements without introducing too much added complexity such as implementation, configuration, and deployment. Since BRD is implemented using a single FIFO queue and a random dropping mechanism, it is simpler than JoBS in complexity perspective. In addition, BRD demonstrated the possibility of providing per-hop differentiated loss guarantees without additional active management such as policing, traffic profiles, or signaling. Moreover, while JoBS shows significant deviations in the desired short timescale, BRD is capable of providing both long and short term performance guarantees.

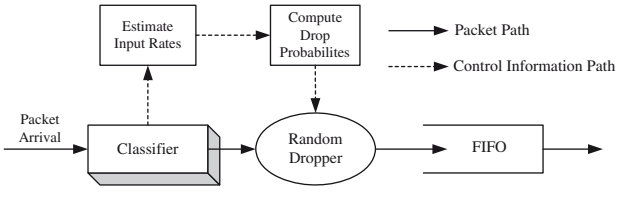


Fig. 1. Bounded Random Drop scheme [6]

The algorithm of the BRD scheme as shown in figure 1 is described as follows:

(1) Arrival rates are estimated using an exponentially weighted moving average with a weighting factor α. For each class i, we use a counter A_i to keep track of the amount of input traffic during each Δt sampling period. At the end of each period, the traffic rates are updated by $r_i = (1-\alpha)r_{i-1} + \alpha A_i/\Delta t$, for class $i = 1, ..., N$.
(2) The target loss probabilities, p_i, $i = 1, ..., N$, are computed based on the r_i's and all counters are reset.
(3) Upon arrival of a packet belonging to class i, the packet will be dropped randomly with the calculated loss probability of class i.

Although BRD has several benefits such as low complexities and good functionalities, we identify that it has some shortcomings (as shown in section 3). Next, we focus on the BRD scheme as it is more relevant to our work.

3 Proposed Scheme

3.1 Motivation

We first analyze the target loss probability in BRD using the equations presented in [6] and show the result in Table 1. For this analysis, we consider a single output link with capacity 10 Mbps. We assume three classes with Constant Bit Rate (CBR) sources. The loss bounds assigned to each class are set to 0.1 for Class 1, 0.2 for Class 2, and none for Class 3. The input rates of the three classes are shown in Table 1.

Table 1. Analysis of target loss probabilities in BRD

Case	Input Rates (Mbps)			Traffic Load	P_1	P_2	P_3
	Class 1	Class 2	Class 3				
1	2	4	2	0.8	0	0	0
2	2	4	4	1	0	0	0
3	4	2	5	1.1	0.0909	0.0909	0.0909
4	4	4	4	1.2	0.1	0.2	0.2
5	5	6	5	1.6	0.1	0.2	0.86
6	**8**	**4**	**1**	**1.3**	**0.1**	**0.3**	**1.0**
7	**10**	**10**	**10**	**3.0**	**0.1**	**0.9**	**1.0**
8	**12**	**1**	**5**	**1.8**	**0.16667**	**1.0**	**1.0**

As we can see, all packets are not discarded for low traffic intensity (see Cases 1 and 2 in Table 1). Case 3 shows that the higher priority classes (Classes 1 and 2) experience the same loss performance as lower priority ones as long as their absolute loss bounds are not violated. Case 4 shows that the higher priority classes (Classes 1 and 2) receive preferential loss treatment only when they are required to avoid violating their own loss bounds. From Cases 5 to 8, we know that the loss bounds of lower priority classes are relaxed first when it is not feasible to satisfy the loss bounds of all traffic classes simultaneously.

Through the analysis of the target loss probability in BRD, we identify that BRD has several shortcomings as the following:

- **Long Queuing Delays** - The BRD scheme does not drop packets before traffic load is greater than one (see Cases 1 and 2 in Table 1). Therefore, it exists occasionally in a case that a buffer is full or reaches a specified threshold, thereby resulting in long packet delays in the buffer.
- **Fairness** - The BRD scheme allows a higher priority class or a few higher priority classes to monopolize the buffer space of the router (see Cases 6-8 in Table 1), preventing other lower priority classes from getting space in the router queue. If the higher priority classes continue to arrive with large traffic volume for a long period of time, the lower priority classes will be highly likely starved. This results in unfair sharing of network resources among the

classes, thereby giving rise to fairness problems. Moreover, 100% drop of the lower priority classes causes all their senders to back off simultaneously - called global synchronization problem.

- **Computational Complexity** - The BRD scheme needs just one comparison for each incoming packet (refer to the BRD algorithm given in the previous section). Moreover, after each sampling period, a few operations including the random number generation are required. They increase as the number of incoming packets increases.
- **Selection of Optimal Values of Parameters** - The BRD scheme depends on the values of its two parameters, sampling period Δt and a weight factor α. Therefore, it is necessary to investigate further the sensitivity of optimal values of Δt and α.

The shortcomings above are investigated for the deterministic traffic. We also examine that the BRD scheme does not support declared level of QoS for the stochastic traffic. Actually, let a size of considered buffer be equal to K. Let p_i be the probability of i packets in the buffer, $i = 0 \ldots K$. Here p_K is the probability of buffer overflow. At this time, all incoming packets are dropped. According to the BRD algorithm, if the number of packets in the buffer is equal to i, where $i = 1 \ldots K-1$, then the drop rate of offered load equals LB_1, where LB_1 is the required level of blocking probability in Class 1. In the case of K packets in the buffer, the blocking probability of incoming packets equals one because the considered buffer is full.

Let us calculate the drop rate, DR, of the higher priority packets

$$DR = LB_1 \sum_{i=0}^{K-1} p_i + p_K \cdot 1 \tag{1}$$

$$= LB_1 \sum_{i=0}^{K} p_i + (1 - LB_1)p_K \tag{2}$$

$$= LB_1 + (1 - LB_1)p_K > LB_1. \tag{3}$$

where $\sum_{i=0}^{K} p_i = 1$ and $LB_1 < 1$. Thus, the drop rate of offered load corresponds to the required level of QoS if and only if $p_K = 0$. It means unlimited buffer, cut-through switching under deterministic traffic, or very low traffic intensity. Otherwise, $DR > LB_1$ and this indirectly implies that the BRD scheme does not support the required level of QoS. It may be concluded that ignored buffer status information is certain to be a shortcoming in QoS-enabled schemes. Accordingly, one of the research objectives of this paper is to eliminate the weaknesses discussed here and to present strong service guarantees with low complexity and high scalability.

3.2 MQDDP

In this subsection we propose a multiple queue management scheme based on differential drop probability, called MQDDP, that supports differentiated loss

guarantees for multiple service classes. MQDDP is based on the properties of the original RED recommended by IETF for the future Internet router. Therefore, it contributes to high interoperability between routers deployed in many commercial networks, and more importantly high reusability of the prevalent RED scheme.

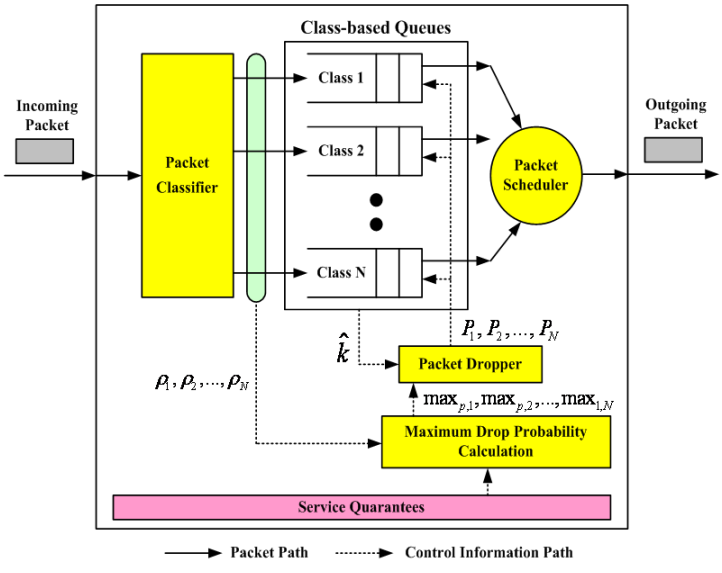


Fig. 2. Architecture of MQDDP

The architecture of the proposed scheme shown in Figure 2 consists of four major modules at routers:

(1) **packet classifier** that can distinguish packets and group them according to their different requirements;
(2) **packet dropper** that determines both of the following: how much queue space should be given for certain kinds of network traffic and which packets should be discarded during congestion;
(3) **packet scheduler** that decides the packet service order so as to meet the bandwidth and delay requirements of different types of traffic; and
(4) **calculation module for the maximum drop probability** that takes the real-time measurement of traffic load and manages the packet dropper.

We first present the basic algorithm of RED on a single queue, and then we extend to the proposed scheme by deriving the drop probability equations based on a queueing model. Since each class-based queue in the proposed scheme is independent and has common properties of RED, the derived equations below are immediately applicable to each queue.

For a queue with RED, incoming packets are dropped with a probability that is an increasing function $d(k)$ of the average queue size k. The average queue size is estimated using an exponentially weighted moving average formula. Let k_0 be the previous average queue size, then $k = (1 - w_q) \cdot k_0 + w_q \cdot \widehat{k}$, where $\widehat{k}$ is the current queue size and w_q is a weight factor, $0 \leq w_q \leq 1$. RED offers three control parameters: maximum drop probability max_p, minimum threshold min_{th}, and maximum threshold max_{th}. It depends on the averaged queue length k with weighted factor w_q to tune RED's dynamics.

For a given resource portion out of common output channel through a predestined scheduling, the service rate of each class queue i, μ_i, is determined. With a certain arrival rate, λ_i, the related system utilization factor, $\rho_i = \lambda_i/\mu_i$, can be assumed. The typical dropping function of class i, $d_i(k)$, in RED is defined by three parameters min_{th}, max_{th} and $max_{p,i}$ as follows:

$$d_i(k) = \begin{cases} 0, & k < \min_{th} \\ \frac{\max_{p,i} \cdot (k - \min_{th})}{\max_{th} - \min_{th}}, & \min_{th} \leq k < \max_{th} \\ 1, & k \geq \max_{th} \end{cases} \tag{4}$$

where $\max_{p,i}$ is maximum drop probability of Class i. Therefore, the drop probability of a packet depending on the dropping function related to each state k can be calculated as follows:

$$P_i = \sum_{k=min_{th}}^{K} \pi_i(k) d_i(k), \quad min_{th} \leq k < K \tag{5}$$

where $\pi_i(k)$ is a probability of k packets in Class i.

We let $\max_{th} = K$ and $\{\lambda_k = \lambda, \mu_k = \mu, \forall k\}$, then the number of packets in the RED queue is actually a birth-death process with the birth rate in state k to $\lambda_i(1 - d_i(k))$ and the death rate to μ_i. The formulas for π_i have been referenced in [9]. Accordingly, the steady-state probability of finding k packets, $\pi_i(k)$, is derived as follows:

$$\pi_i(k) = \pi_i(0) \rho_i^k \prod_{l=0}^{k-1} (1 - d_i(l)) \tag{6}$$

where $\pi_i(0) = \left[\sum_{k=0}^{K} \rho_i^k \prod_{l=0}^{k-1} (1 - d_i(l)) \right]^{-1}$.

Next, we should determine the values of $max_{p,i}$ as the dominant control parameter. They can be easily calculated from the accurate formulation in Eq. (5) when the required drop probability in a class is given under the offered load ρ_i. For simplicity, we consider the values of $max_{p,i}$ for loss probability of 0.01, 0.02, and 0.04. shown in Figure 3. These values can be configured into the max_p-by-ρ reference table. The algorithm of the MQDDP scheme is described as follows:

(1) Traffic loads and current queue sizes are monitored on the tth sampling interval.

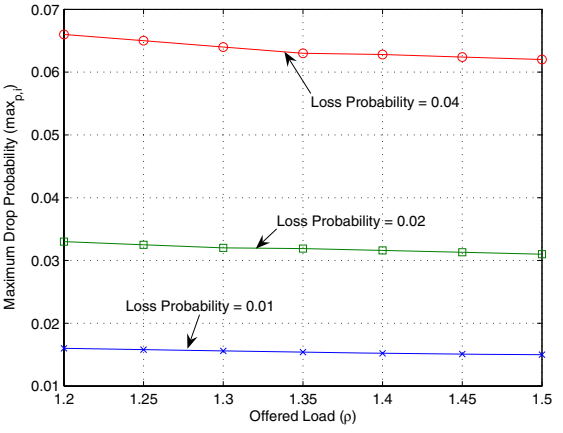


Fig. 3. $max_{p,i}$ versus offered load

(2) At the end of each period, the values of $max_{p,i}$ are obtained from the max_p-by-ρ reference table for loss requirements and average queue sizes are estimated by k.
(3) $d_i(k)$ and $\pi_i(k)$ are computed by Eq. (4) and Eq. (6), respectively.
(4) The loss probabilities, P_i, $i = 1, ..., N$, are calculated by Eq. (5).
(5) Upon arrival of a packet belonging to class i, the packet will be dropped randomly with the calculated loss probability of class i.

It is certain that the packet drop rate of the proposed method is a continuous function of outgoing channel capacity. Hence, we can obtain more reasonably the required level of packets blocking probability by rescheduling the outgoing channel bandwidth. Thus, the proposed scheme provides absolute differentiated loss guarantees. For example, if $max_{th} = 64, min_{th} = 2, max_p = 0.5, w_q = 1$, Poisson flow of incoming packets, and the required level of packets loss is equal to 10%, then the capacity of outgoing channel should be greater than 90% of the offered load rate. If the required level of packets loss is 5%, then we need to increase the outgoing channel bandwidth up to the offered load rate.

4 Comparison of BRD and MQDDP

In this section, we compare the proposed scheme with BRD for highly offered loads, which mean an offered loads essentially exceed the outgoing channel capacity. In other words, QoS support for all classes for the traffic is impossible. The schemes for differentiated loss guarantees keep higher priority packets and drop other traffic. Without loss of generality, we consider a single class of traffic. The criterion for comparison of BRD and MQDDP is an attainable level of packet drop rate.

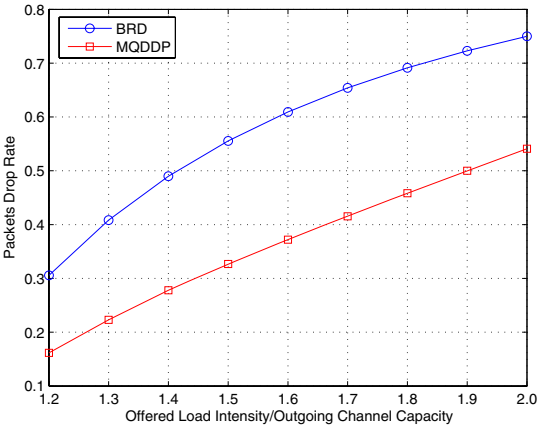


Fig. 4. Comparison of BRD and MQDDP under very high offered loads

An offered load is assumed to be the Poisson process with rate kC, where C is the outgoing channel capacity, k is a constant, and $1.2 \leq k \leq 2$. Let the maximal number of packets in the buffer be 64. The parameters of the proposed scheme are given by $max_{th} = 64$, $min_{th} = 2$, $max_p = 0.5$, and $w_q = 1$.

According to the BRD algorithm, a part of the offered load is randomly dropped. In the case of highly offered loads, the probability of incoming packet dropping equals $1 - \frac{1}{k}$. The packet drop rate for BRD scheme is

$$1 - \frac{1 - p_{BO}}{k} \tag{7}$$

where p_{BO} is the probability of buffer overflow. It has been defined by the blocking probability formula of $M/M/1/64$ queuing system.

By assigning different offered load rates, we can get packet drop rates for each scheme as shown in Figure 4. It may be concluded that the proposed scheme has better performance in terms of packet drop rate. For instance, if the ratio of offered load intensity to outgoing channel capacity equals 1.5, the proposed scheme reduces the packets drop rate about 40%.

5 Conclusion

In this paper, we investigated a technique of absolute differentiated loss guarantees. While BRD has several benefits such as low complexities and good functionalities, we identify that it has some shortcomings such as low throughput, long queuing delays, and selection problem of optimal values of parameters. Specifically, this paper focused on low throughput of BRD's shortcomings. It was shown that ignored buffer status information causes inevitable weakness in QoS-enabled schemes. The proposed scheme is based on the properties of the

original RED with a mechanism for absolute loss guarantees, eliminates the mentioned weaknesses as shown the comparison between MQDDP and BRD in case of high traffic intensity, and is shown to have the better performance in terms of packet drop rate. In the future, it is reasonable to investigate an optimal choice of several parameters in the proposed scheme with differentiated services.

References

1. J. Aweya, M. Ouellette, and D. Y. Montuno, "Weighted Proportional Loss Rate Differentiation of TCP Traffic," *Int. J. Network Mgmt*, pp. 257-272, 2004.
2. Y. Chen, M. Hamdi, D. Tsang, and C. Qiao, "Proportional QoS provision: a uniform and practical solution," In Proc. of ICC 2002, pp. 2363-2366, 2002.
3. C. Dovrolis and P. Ramanthan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model," *IEEE Network*, 13(5):26-34, 1999.
4. C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," In *Proc. of ACM SIGCOMM*, pp. 109-120, 1999.
5. C. Dovrolis and P. Ramanathann, "Proportional differentiated services, part II: loss rate differentiation and packet dropping," In *Proc. of IWQoS*, pp. 52-61, June 2000.
6. Y. Huang and R. Guerin, "A simple FIFO-based scheme for differentiated loss guarantees," In *Proc. of IWQoS*, pp. 96-105, 2004.
7. J. Liebeherr and N. Christin, "JoBS: joint buffer management and scheduling for differentiated services," In *Proc. of IWQoS*, pp. 404-418, 2001.
8. J-S Li and H-C Lai, "Providing proportional differentiated services using PLQ," In *Proc. of Globecom*, pp. 2280-2284, 2001.
9. Vladimir V. Shakhov, Jahwan Koo and Hyunseung Choo, "On Modelling Reliability in RED Gateways," In *Proc. of ICCS*, pp. 948-951, 2006.
10. J. Zeng and N. Ansari, "An enhanced dropping scheme for proportional differentiated services," In *Proc. of ICC*, pp. 1897-1901, 2003.

Dynamic Location Management Scheme Using Agent in a Ubiquitous IP-Based Network

Soo-Young Shin[1], Soo-Hyun Park[1,*], Byeong-Hwa Jung[1], and Chang-Hwa Kim[2]

[1] School of Business IT, Kookmin University,
861-1, Chongnung-dong. Sungbuk-gu, Seoul, Postfach 136-702, Korea
[1]{sy-shin, shpark21, bhjung}@kookmin.ac.kr
[2] Kangnung University
chkim@kangnung.ac.kr

Abstract. IP-based IMT Network Platform(IP^2) is an ubiquitous platform supporting mobility using two step IP address mode converting – IP host address(IPha) and IP routing address(IPra) – in a backbone network. Mobile Node(MN) in IP^2 maintains its state as either active or dormant state, which is transferred to active state through Paging process when communication is required. In this paper, we proposed a Paging method using agent to resolve the problem of the conventional Paging method which transmits the Paging messages to all cells in Location Area(LA) resulting in the excessive use of network resources. Performance evaluation of the proposed method using NS-2 showed that the usage of network resources becomes more efficient by reducing Paging loads, especially under the condition of increased nodes.

1 Introduction

As all countries of the world are trying to construct ubiquitous infra, the present 3rd generation network is ready to evolve into 4th generation network, in which IP is a backbone network. Since the advent of ubiquitous age will bring about an explosive increase of large multimedia traffics, ubiquitous networks should provide wideband seamless mobility and stabilized services simultaneously. ITU-R has presented that every telecommunication networks would be changed to IP based networks to transmit large multimedia traffics with stability. [1]

NTT DoCoMo has suggested IP based IMT Network Platform (IP^2) [2],[3] as the next generation All-IP mobile network structure taking into account the increased multimedia traffics and IP technologies. The basic structure of IP^2 is categorized into three classes, which are IP-Backbone(IP-BB), Network Control Platform(NCPF) and Service Support Platform(SSPF) respectively. Among Mobility Management functions of NCPF, Local Manager(LM) plays an important role to manage information of all MN locations and enable the MN in Dormant state to change its state to Active after conducting Paging process. During the Paging process, LM makes flooding the Paging packets to every node in LA (Location Area) to which MN is belong. At the moment, the overall network efficiency is significantly lowered to

* Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 491 – 500, 2006.
© Springer-Verlag Berlin Heidelberg 2006

process the Paging of one MN resulting in the waste of network resources. To resolve this problem, we suggest using an agent server for Paging. The agent server for Paging will have the detailed information of MN location so that it mitigates the waste of network resources by conducting selective Paging at the request of Paging.

In Chapter 2, the procedures of the conventional IP^2 network Paging and the proposed Paging mechanism with agent are described. The mobility of IP^2 network is described in Chapter 3 respectively. Paging Cost in IP^2 network with agent server is described in Chapter 4 and Network simulation results in Chapter 5 and conclusions in Chapter 6 are presented.

2 Paging and Role of Agent Server in IP^2 Network

IP-BB of IP^2, which is a backbone network of ubiquitous network, has been designed to support mobility. It uses two types of IP address - IPha and IPra - to support free mobility of MN. IPha is used between MN and Access Router(AR) to identify MNs within the cell of AR and IPra is used to transmit packets within IP-BB. Network structure of IP-BB has imported the feature of cellular networks, which manage the routing in combination with the area. MN in Dormant state is managed apart by LM so that it is not needed to send signals updating the location information of MN whenever it moves. Besides, there is no necessity for managing and keeping the information of MN in Dormant state in Cache for Source Terminal(CST) and Cache for Destination Terminal(CDT). As a result, the required network resources are reduced. [4] Active state of MN, which includes the state of data transfer or standby, can be changed to dormant state according to timeout value. Dormant is a suspending state in its communication. These two states exchange themselves mutually. Figure 1 shows a state transition diagram. The arrow (1) indicates sending activation message to AR from MN in Dormant state. The arrow (2) is the case that LM sends Paging message to MN in dormant state. The arrow (3) is the case that MN receives time-expiration message from AR. Paging means the procedures conducted by LM for state transfer from Active state to dormant state.

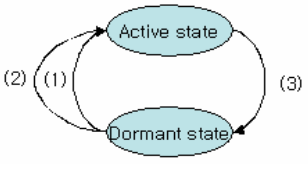


Fig. 1. State transition diagram

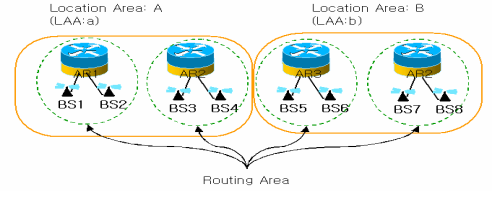


Fig. 2. Regional Classes of Location Area and Routing Area

LM manages the location information of all MNs within IP^2 networks and changes MN in Dormant state to Active state by Paging procedure. Besides, it identifies the area, at which MN is located, as Location Area Address(LAA) and stores it in Location Management Table. RM assigns IPra to active MN and manages the information of routing addresses. AR changes IPha in the packets into IPra and transmits the packets to the network. AR has CST, which stores IPha and IPra of

source MN, and CDT, which stores IPha and IPra of destination MN [5]. LM manages the information of MN locations in terms of LA. And RM manages the information of MN locations concerning with RA. Figure 2 shows the regional classes of LA and RA.

Since there is no concept of dormant state in Mobile IP, the location information should be updated every time MN moves even if there is no communication. Consequently, the waste of power and network resources becomes significant [6]. In IP^2, however, the problem of wasting power and network resources are mitigated because LM maintains a dormant state of MN. The new problem of using LM is that the excessive use of network resources is caused when the state is changed from Dormant to Active through the Paging procedure, by which the messages are flooding to all MN within LA. To resolve this new problem, we proposed to use an agent server for Paging. Since the agent server has more detailed information of MN location than LM does, it can obtain the information of a specific path while the Paging messages transmitted from LM pass through the agent server. As a result, only AR and Base Station(BS), to which IPha of MN belongs, would be transmitted the Paging packets and the network resource usage becomes more efficient.

Figure 3 shows the procedure of registering detailed information of MN location on an agent server. The location register is the procedure of registering MN's new location in case that MN joins a certain LA first or moves to other LA. In general, the location registration is conducted during MN's moving to other LA. The sequence of the location register is as follows. 1) *MN#1* in Dormant state detects its location has changed. 2) *MN#1* transmits a location registering message to *BS3* of *AR2*. 3) *AR2* transmits a location update message to LM. 4) Entry of *MN#1* in the location management table of LM is updated. 5) The location registering message sent by *AR2* is transmitted to the agent server. 6) Entry of *MN#1* in the agent table of the agent server is updated. 7) The agent server transmits a location updating Ack to *AR2*. 8) *AR2* transmits Ack about the location registering message to *MN#1*. A location registering procedure is finished in this way. Figure 4 shows the Paging procedure in general IP^2 networks without agent servers. It can be noticed that the resources are

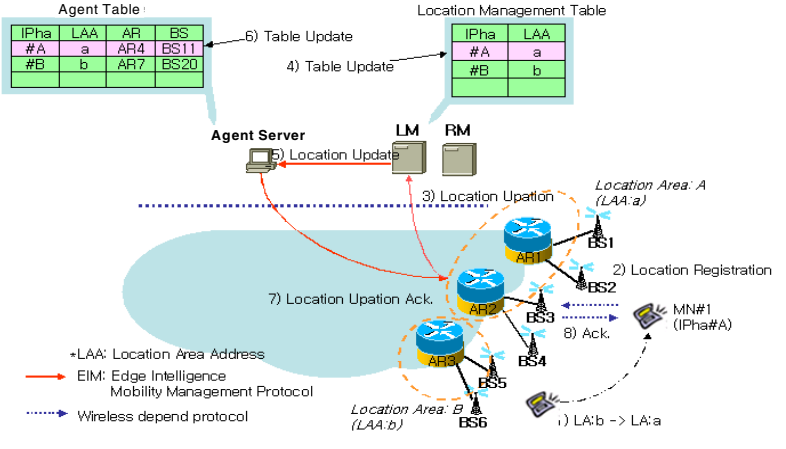


Fig. 3. Procedure of location registering

being wasted since *IPha#A* is transmitting to every MN in LAA, to which *IPha#A* is subjected, through *AR1* and *AR2*. RM transmits a Paging trigger to LM. LM searches for LAA of the requested IPha through making a search for IPha in its location management table. 2) LM asks ARs in the area for Paging using LAA. 3) Each AR starts Paging through AP which uses Layer 2 Paging signals. 4) *MN#1* replies by transmitting a Paging Ack. And then, MN conducts an activation procedure.

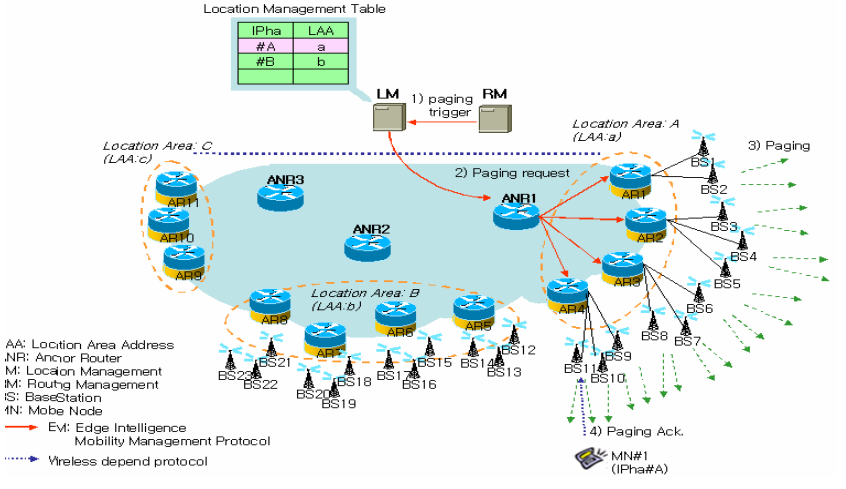


Fig. 4. Paging Procedure in general IP^2 network

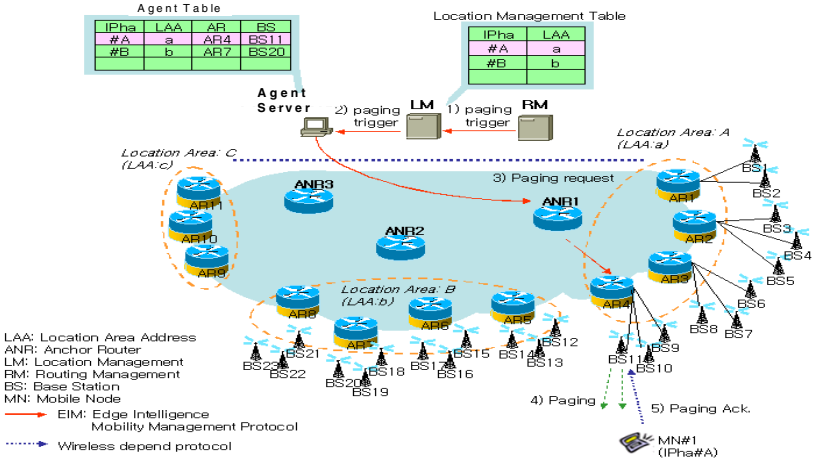


Fig. 5. Paging procedure in IP^2 network with agent server

Figure 5 shows the Paging procedure in IP^2 network environment with agent server. Since the agent server has the exact information of AR and BS, at which *MN#1* is placed, it is not needed to transmit Paging messages to full area of LAA.

Consequently, the efficiency of network resources usage can be significantly improved comparing to the case of Figure 4. *RM* transmits a Paging trigger to LM. LM searches LAA of the requested IPha by making a search for IPha(in figure 5, *IPha#A*) in its location management table. 2) LM transmits the Paging trigger to agent server. The agent server start searching procedure for the exact AR and BS within the entry of IPha in its agent tables. As a result, the agent server finds that MN has the *IPha#A* belong to *AR4* and *BS11*. 3) The agent server asks *AR4* for Paging using the information found in agent table. 4) *BS11* in *AR4* starts Paging through AP which uses Layer 2 Paging signals. 5) *MN#1* replies by sending a Paging Ack. Then MN begins the activation procedure.

Here is the figure 6 which shows the message flow diagram for location register in IP^2 network environment without agent server. When the mobile node *MT#C* enters to the area of BS, *MT#C* receives advertising message from BS. Then location register message for *MT#C* arrives to LM through AR and LM sends back ACK message to *MT#C* notifying to be registered to LM. The lower part under a dotted line designates the process for *MT#M* to send data packet to *MT#C*.

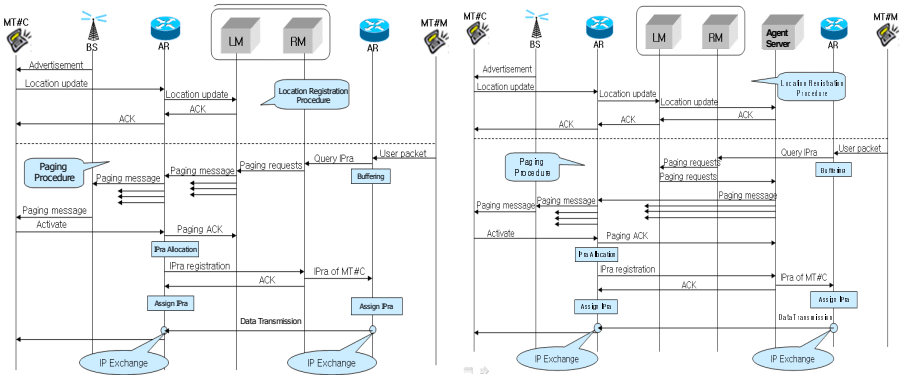


Fig. 6. The message flow in IP^2 network (without agent server(left) / with agent server(right))

After AR receives the data packet for *MT#M* transmitting to *MT#C*, it finds some IPra information about the destination of data packet at it's own DST. After assuring that AR does not have IPra for destination node, *MT#C*, it requests IPra for *MT#C* to RM. The absence of IPra for *MT#C* causes Paging request to LM to find the exact location of destination node, *MT#C*. After LM searches LA for IPha of destination node, it floods Paging message to LA having *MT#C*. After *MT#C* receives this Paging message, AR assigns new IPra to *MT#C* through Activation procedure and registers new IPra to RM. At the next step, RM forwards IPra of *MT#C* to source AR which *MT#M* is belonged. The data packets buffered at source AR are send to *MT#C* following after substitution IPra for destination address of packets.

Figure 7 explains the message flow diagram for location register in IP^2 network environment with agent server. At the location register process, LM delivers message for location information to agent server. In the course of Paging process, LM sends Paging request message to agent server. In other words, the agent server acts for Paging mechanism.

3 Mobility in IP^2 Network

In the scenario which is shown in Figure 8, there are three IP^2 domains and MN#M in IP^2 domain 3 is going to transmit packets to *MN#C* in domain 1. *MN#M* is in activation state and IPra is already allocated to AR and RM to which *MN#M* is belonged. However, *MN#C* is in dormant state. Only its location information is registered to LM, to which *MN#C* is belonged and IPra is not allocated to AR and RM. Mobility management procedure is as follows.

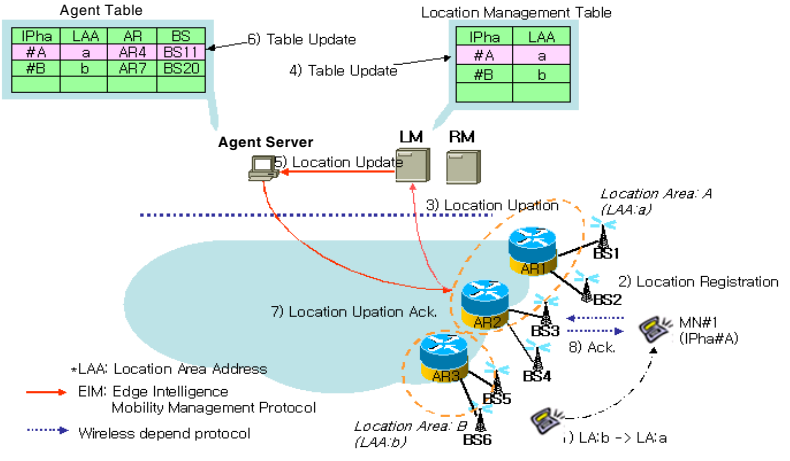


Fig. 8. Mobility scenario of IP^2 network with Agent Server

In case of 1), sender *MN#M* in IP^2 domain 3 transmits the first packet for communication to receiver *MN#C* in AR1 of IP^2 domain 1. Then AR6 makes a search for IPra corresponds to the destination IPha of *MN#C* in its CDT. Since IPha of *MN#C* is not listed in the table directory, however, AR6 asks RM in MM3 for IPra with regard to IPha of destination *MN#C*. If *AR6* finds there is no IPra in RM in *MM3* also, *AR6* checks on if exists in its LM in Dormant state. If *AR6* finds it doesn't exist, *AR6* asks MMs in other domain for IPha information. After *AR6* finds the IPra does exist in *MM1* of domain 1 in dormant state, Paging procedure is conducted in LM of *MM1*. Then, LM asks agent servers to conduct Paging to the location where *MN#C* is located. Through the Paging procedure, *MN#C* conducts activation procedure and *AR1* allocates IPra and registers it to RM of *MM2*. This IPra information is transmitted to *AR6* in IP^2 domain 3. Then, IPra in CDT of *AR6* is updated and the packet changes its destination address into IPra and is transmitted to the destination.

In case of 2), *MN#C*, which is communicating with *MN#M*, moves from LAA:a to LAA:b region. During this stage, the location register procedure should be conducted because the handover is a guarantee to mobility. Through the location register procedure, LM, a location information of *MN#C* in agent servers and the information of CDT in AR, at which every communicating MN is located, are updated. The situation of 3) is that *MN#C* is moving within LAA:b region, allocated new IPra from AR and registers it to RM. Then, RM updates CDT information of AR, at which all

MN communicating with *MN#C* is located. Lastly the case of 4) is that *MN#C* moves from domain 1 to domain 2. *MN#C* conducts a location register procedure and registers its IPha to LM of *MM3*. The location information of *MN#C* is deleted in LM and RM of *MM1*. Then, IPra is allocated to *AR4* and RM of *MM2* CDT information of *AR4*, at which all MNs communicating with *MN#C* are located, is updated.

4 Paging Cost in IP^2 Network with Agent Server

In IP^2 ubiquitous back-bone network, we can apply Blanket paging scheme to evaluate Paging cost. his scheme is to send Paging message to all the cell which belongs to Location Area. With all making a delivery of Paging message to all cell cause big traffic load, it has predominant the advantage of prompt response. The main purpose of considering IP^2 network as a lot of Paging area is to reduce network traffic as well as to improve network performance This Paging mechanism is called sequential paging scheme. The main concept proposed in this paper is to transmit Paging signal to MN more accurately by using Agent server. Measuring cost of sequential paging scheme is the derivative following equation.

$$C = L + \sim : D \tag{1}$$

In this equation, C, L, D and $\sim$ mean total paging cost, cost of paging load, paging delay cost and delay factor. Added to this, L has significant meaning for the total summation of traffic loads in all cells till Paging success. When we concluded a little earlier that there is the total cost of traffic loads in all cells till Paging success, this was equivalent to saying that this total cost (for L) has the meaning of Paging failure to success to find the location of a lost MN in a sense. Of course, Paging delays arising from Paging failure are extended to cover the notion of D. When we apply the equation (1) to Blanket paging scheme, L is the traffic load for all cells of LA and D becomes insignificant. While there can be no doubt that L and D are insignificant as agent allows LM to know MN's true positions, there must be considerable doubt that there are any errors in case of using Agent server. The last location of MN is stored at agent when MN is deactivated and the location of MN is not be updated until only at certain points, such as when MN is activated. However, We can not find MN once MN moves to any other area(LA) in this situation. The Blanket paging scheme is used to search for the lost MN at this point of time.

Let $C_{blanket}$ be a Blanket paging cost and C_{agent} Paging cost with agent server. We can express the Paging cost in the form of equation (2).

$$C_{agent} = \mathbf{b} : C_{blanket} \tag{2}$$

Notice in this case that $\mathbf{b}$ is probability for MN to move another area after deactivation. There are 2 states for MN and if P_h is the probability to move from this cell to neighbor cell, $\mathbf{b}$ is defined as expression (3). Also we can obtain the following equation (4) for P_h. [9]

$$\mathbf{b} = \frac{1}{2} : p_h \tag{3}$$

$$C_{proxy} = \frac{1}{2} \cdot \left[\frac{1 - e^{-\alpha}(1-\alpha)}{2\alpha} - \frac{\alpha}{2} \right] \cdot \int_{\alpha}^{+\infty} \frac{e^{-x}}{x} dx \right] \cdot C_{blanket} \quad (4)$$

Therefore C_{agent} is specified as follows (5).

$$C_{proxy} = \frac{1}{2} \cdot \left[\frac{1 - e^{-\alpha}(1-\alpha)}{2\alpha} - \frac{\alpha}{2} \right] \cdot \int_{\alpha}^{+\infty} \frac{e^{-x}}{x} dx \right] \cdot C_{blanket} \quad (5)$$

5 Simulation

For the performance analysis of the proposed agent server, which aims to more efficient Paging in IP^2 environment, we have conducted a series of simulation using NS-2 (Network Simulator 2). There is no conventional package perfectly supporting IP^2 environment in NS-2. CIMS(Columbia IP Micro-Mobility Suite), which is a NS-2 extension, supports micro mobility-related Cellular IP, Hawaii, Hierarchical Mobile IP [10]. Among these three packages, Cellular IP has some features of IP^2. In this package, IP is used to discriminate hosts and MN has active / dormant state. Besides, MN has features supporting the location management and Paging functions.

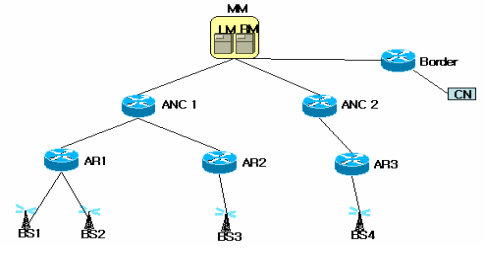


Fig. 9. Topology graph of the simulation model

In this paper, we have added LM of MM in IP^2 and implemented the proposed agent server and a Paging mechanism of IP^2. For comparison and verification of Paging loads, we have configured an environment with variable size of LA and variable number of MNs. Each LA has four BS and there are two LAs (eight BSs) in the configured topology. Figure 9 shows the topology graph of the simulation model.

As shown in Figure 10, the proposed Paging technique with agent servers has presented more decreased Paging loads comparing to conventional Paging technique.

Table 1 shows exact quantitative values and performance improvement ratio corresponding to various Paging intervals. When mobility is increased (=an increased Paging interval), the performance improvement ratio has remarkably improved to 43.57 %. This value is almost double comparing to the case of single LA with 2 BS.

Figure 11 shows the Paging loads in case that the number of MNs are increased by four-fold. Table 2 shows quantitative indices of Figure 8 with regards to various Paging intervals. When mobility is increased, the performance improvement rate rises up to 42.26 % comparing to the case of conventional Paging. Comparing with the case of 2 MN, the rate is 3~6 %. Table 3 shows the performance improvement rates in case that the number of BS and MN is fixed.

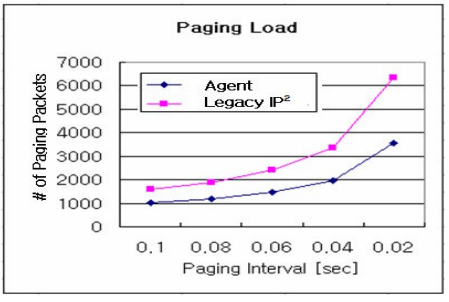


Fig. 10. Paging Load (BS:8, MN:2)

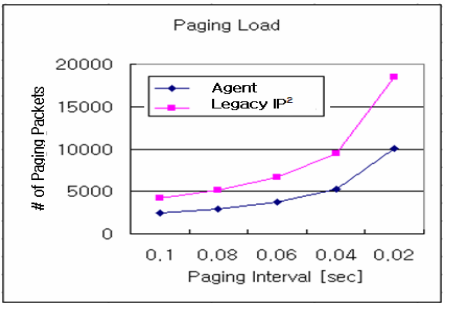


Fig. 11. Paging Load (BS:8, MN:8)

Table 1. Paging Load (BS:8, MN:2)

Interval (ms)	100	80	60	40	20
Legacy (#of Paging)	1,591	1,903	2,423	3,372	6,336
Agent (#of Paging)	1,021	1,198	1,469	1,980	3,576
Performance improvement ratio(%)	35.83	37.05	39.38	41.29	43.57

Table 2. Paging Load (BS:8, MN:8)

Interval (ms)	100	80	60	40	20
Normal (#of Paging)	4,205	5,115	6,659	9,527	18,446
Agent (#of Paging)	2,465	2,951	3,777	5,317	10,098
Performance improvement ratio(%)	41.38	42.31	43.28	44.2	45.26

Table 3. Paging Load

Interval (ms)	100	80	60	40	20
BS:8-MN:8	41.38	42.31	43.28	44.2	45.26
BS:8-MN:2	35.83	37.05	39.38	41.29	43.57
BS:4-MN:8	18.87	19.53	20.14	20.69	19.86
BS:4-MN:2	19.38	20.27	20.72	21.18	21.68

Simulation results shows that the performance of the proposed technique increases as the area of LA becomes larger, as the number of BS increases and as the number of MN increases. The maximum improvement rate was 45 %. Since the area of LA and the MN node number is on an increasing trend in real world backbone networks, the proposed method is expected to be more effective one, especially in the next generation network environment, in which the number of ubiquitous terminals would be increased rapidly.

6 Conclusion

As a 4G (Generation) network architecture, IP^2 is the architecture which is based on a new concept and proposed to cope with large multimedia traffics generated by mobile communications in the near future. MN in IP^2 has either dormant or active state. Dormant state is a suspended state of communication and the state has to be changed into active state through Paging procedures for restarting communications. During this Paging procedure, signaling messages have to be transmitted to all terminals within LA. Consequently, sharing resources of network is used excessively since many MNs shall ask communication during dormant state or handover between LA. To mitigate this shortcoming of the Paging feature of LM in IP^2, a scheme of using agent servers is

proposed in this paper. Since the agent servers for Paging procedures are keeping the information of detailed location of MN, the Paging message can be transmitted to the exact location of LA in case of a Paging request from LM. Consequently, the sharing resources of the network can be used efficiently. NS-2 simulation results for the performance evaluation of the proposed technique have shown that the Paging efficiency can be improved by up to 45 %. Since the proposed technique does not support any registration procedure according to handover in case that dormant MN moves to other LA, however, additional supplement of this feature is needed.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

Reference

1. ITU-R Draft Recommendation, “Vision, framework and overall objectives of the future development of IMT-2000 and systems beyond IMT 2000,” November 2002
2. H.Yumiba,et al., “ IP-based IMT Network Platform,” IEEE Personal Communication Magazine, Vol. 8, No. 5, pp. 18-23, October 2001.
3. K. Imai, M. Yabusaki, and T. Ihara, “IP^2 Architecture towards Mobile Net and Internet Convergence,” WTC2002, September 2002.
4. Atsushi IWASAKI, Takatoshi OKAGAWA, “Scalability Evaluation of IP-based IMT Network Platform.,” IEICE Technical Report, Vol. 104, No. 184, NS2004-69, pp. 5-8, July 2004.
5. Takatoshi Okagawa, et al., “Proposed Mobility Management for IP-based IMT Network Platform,” IEICE Transactions on Communications 2005, Vol. E88-B , No. 7, pp. 2726~2734, 2005
6. Katsutoshi Nishida, et al., “Implementation and Evaluation of a Network-Controlled Mobility Management Protocol (IP2MM): Performance Evaluation Compared with Mobile Ipv6,” Wireless Communications and Networking Conference, 2005 IEEE, Vol. 3, pp. 1402 – 1408, March 2005
7. C. Rose and R. Yates, “Ensemble polling strategies for increased paging capacity in mobile communication Networks,” ACM-Baltzer J. Wireless Networks, vol. 3, no. 2, pp. 159-177, 1997
8. Dong-Jun Lee, et al., “Intelligent Paging Strategy based on Location Probability of Mobile Station and Paging Load Distribution in Mobile Communication Networks,” 2004 IEEE International Conference on, Vol. 1, pp. 128 - 132. June 2004
9. E. Del Re, Senior Member, IEEE, R. Fantacci, G. Giambene, “Handover and Dynamic Channel Allocation,” IEEE Transactions ov vehicular technology, VOL. 44, NO. 2, 1995
10. A. T. Campbell, Gomez, J., Kim, S., Turanyi, Z., Wan, C-Y. and A, Valko “Comparison of IP Micro-Mobility Protocols,” IEEE Wireless Communications Magazine, Vol. 9, No. 1, February 2002.

Detecting and Identifying Network Anomalies by Component Analysis

Le The Quyen[1], Marat Zhanikeev[1], and Yoshiaki Tanaka[1,2]

[1] Global Information and Telecommunication Institute, Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan
[2] Advanced Research Institute for Science and Engineering, Waseda University
17 Kikuicho, Shinjuku-ku, Tokyo, 162-0044 Japan
quyenlt@fuji.waseda.jp, maratishe@aoni.waseda.jp,
ytanaka@waseda.jp

Abstract. Many research works address detection and identification of network anomalies using traffic analysis. This paper considers large topologies, such as those of an ISP, with traffic analysis performed on multiple links simultaneously. This is made possible by using a combination of simple online traffic parameters and specific data from headers of selective packets. Even though large networks may have many network links and a lot of traffic, the analysis is simplified with the usage of Principal Component Analysis (PCA) subspace method. The proposed method proves that aggregation of such traffic profiles on large topologies allows identification of a certain set of anomalies with high level of certainty.

Keywords: Network anomalies, Anomaly detection, Anomaly identification, Principal component analysis, Traffic analysis.

1 Introduction

Nowadays, computer networks and traffic running through them are increasing at a high pace to meet users' requirement. Beside the major proportion of productive traffic, there are many types of network anomalies. The prominent point of all network anomalies is that they generate abnormal changes in traffic features such as bandwidth, load, and other traffic metrics. In this paper, we concentrate on large network topologies connecting many networks by multiple links. Controlling network anomalies in this scope requires collecting traffic and offline processing data on all network links simultaneously. In our research, we propose to use simple link utilization metrics, i.e. bandwidth (bps), load (packet per second), and counters for selective packets to detect and diagnose network anomalies. This paper applies component analysis and uses subspace method to detect abnormal exhibitions in each link metric. This discovery about anomalous features in network traffic allows us to detect and identify them in a timely manner. The efficiency of this method depends on the data sampling rate which contains the detailed level of network traffic.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 501 – 504, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Traffic and Network Model

Within the scope of this paper, we only address a certain set of network-centric anomalies including: AlphaFlow, Scanning, Neptune, Network outage, and FlashCrowds. Introduction about these anomalies can be found in Wikipedia webpage [1]. Regarding connection establishment behaviour, we noticed that network-centric anomalies fall into 3 categories: point-to-point, point-to-multipoint, and multipoint-to-point. Therefore, in order to decide if an anomaly belongs to any of the 3 categories, we count the number of packets with distinguished source-sockets (address and port) and the number of packets with distinguished destination-sockets appearing on each link. As shown in Table 1, by analyzing link bandwidth, load, the number of distinguished source-sockets and destination sockets to detect abnormal change, we can get significant information to identify network anomalies.

We use OPNET Modeler to simulate a backbone network with 7 router nodes and 8 physical links as shown in Fig. 1. We use most of the common network services to provide normal traffic in this topology including: web browsing, email, database access, file transfer, and voice over IP calls. In order to inspect and to verify the efficiency of the proposed method, we subjectively insert 6 anomalies including UDPFlood, Network outage, FlashCrowds, Portsweep, Neptune and IGMPFlood into the network at various specific times. These anomalies are recreated from packet trace files generated by attacks in real network environment, and then imported and regenerated in OPNET. We run the simulation for 1 day and collect values for each metric from all links at 2 sampling intervals: 30 seconds and 5 minutes.

Table 1. Distribution of values in different features of network anomalies

Anomaly	BW	Load	S-socket	D-socket	Timespan
AlphaFlow	large	large			short to long
Spoof or distributed flooding, FlashCrowds	large	large	large		long
Network outage	small	small	small		long
Neptune		large	large		long
Scanning		large	large	large	short

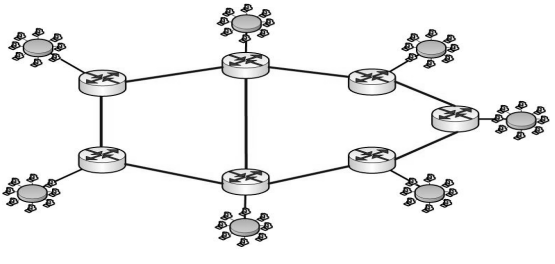


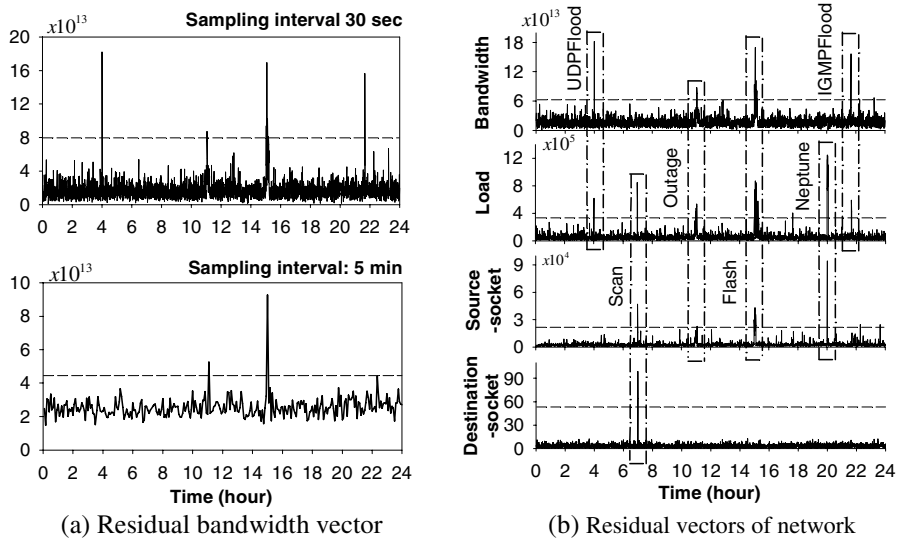
Fig. 1. Network topology

3 Analysis by PCA Subspace Method

In order to learn the current state of network traffic, we create the state matrix for all links where the rows represent sequential timeline for data collection and the columns represent traffic links. Then, the dimensionality of network traffic state is decided by the number of traffic links, which is usually quite large. Our intention is to detect abnormal values or spikes in metric values of network traffic so we apply PCA subspace method. This method is also used in some other researches for anomaly detection purpose [2], [3]. Detailed explanation can be found in our previous research [4].

PCA is used to explore the intrinsic data dimensionality, and transform the given datapoints into a new space with new coordinates. The dataset will be transformed into a new space generated by PCA where principal components (coordinates) build the traffic normal behaviour and residual components build the anomalous behaviour. Therefore, the PCA space is divided into 2 subspaces: normal (from principal components) and anomalous (from residual components). By calculating the residual vector (constituted by values on all coordinates of anomalous subspace), we can detect anomalies in certain network traffic metric.

Fig. 2(a) shows the detection results by subspace method on the 30-second and 5-minute sampling interval dataset of traffic bandwidth. The thresholds to decide what peaks are anomalies are calculated based on the research on residual components in [5]. According to Fig. 2(a), only the 30-second sampling interval allows trustworthy detection as it successfully detects 4 inserted anomalies. The reason is that PCA is an energy-based analysis, and that it can only detect abnormal variances which are strong enough to create significant change.



(a) Residual bandwidth vector at different sampling intervals

(b) Residual vectors of network link metrics

Fig. 2. PCA subspace method detection results

The duration of the inserted UDPFlood and IGMPFlood is 1 minute and 30 seconds respectively, so all features in traffic are smoothed out at the end of a 5-minute interval. In contrast, the duration of Network outage and FlashCrowds anomalies are both more than 10 minutes, which ensures that the measurement interval will result in an anomalous reading. This supports the notion that the higher the traffic data sampling rate, the better the detection result.

4 Identifying Anomalies

According to the discussion in Section 2, the anomalous behaviour of distinct types of anomalies has different signatures in each of the four main metrics: bandwidth, load, the number of distinct source sockets and distinct destination sockets. Therefore, we also apply subspace method to the other 3 parameters of all links to detect anomalous patterns in each individual traffic feature. The results are put together for simultaneous analysis as shown in Fig. 2(b). We see that network anomalies exhibit anomalous behaviour differently in each of the 4 traffic metrics. The detection results match the initial comment in Table 1. In case of network outage, even though when it occurs, most of the metrics decrease abnormally but in PCA analysis, such exhibition is still considered as data variance so it will create spikes in residual vectors. This makes the detection result of network outage similar to that of FlashCrowds. Besides, when there is a failure in a part of the topology, network nodes always tend to find substitutive resource to use, so the variance of traffic metrics is not large enough to create major spikes.

5 Conclusion

In this paper, PCA subspace method is applied to detect and to identify network anomalies based on link traffic analysis. With the traffic obtained through simulation experiments, the efficiency of the proposed method in detecting network anomalies is proved. We also address the issue of sensitivity of the method that depends on the rate of traffic sampling. Since PCA is an energy-based analysis, the method is more accurate when anomaly's energy (variability) is comparatively high. It is left for further study to verify the method using traffic from real network environments which allow us to add parametric substantiality to the proposed method.

References

1. Wikipedia webpage link: http://en.wikipedia.org/wiki/Main_Page
2. Wang, W., Battiti, R.: Identifying Intrusions in Computer Networks Based on Principal Component Analysis, Technical Report DIT-05-084, University of Trento (2005)
3. Lakhina, A., Crovella, M., Diot, C.: Diagnosing Network-Wide Traffic Anomalies, Proc. ACM SIGCOMM, Portland, USA (Aug. 2004) 4-6
4. Le, T. Q., Zhanikeev, M., Tanaka, Y.: Component Analysis in Traffic and Detection of Anomalies, IEICE Technical Report on Telecommunication Management, No.TM2005-66 (March 2006) 61-66
5. Jackson, J. E., Mudholkar, G. S.: Control Procedures for Residuals Associated with Principal Component Analysis, Technometrics (1979) 341-349

Preventive Congestion Control Mechanisms in ATM Based MPLS on BcN: Detection and Control Mechanisms for a Slim Chance of Label Switched Path

Chulsoo Kim, Taewan Kim, Jin hyuk Son, and Sang Ho Ahn

Computer Science Inje University 621-749, Obang-dong, Gimhae, Gyeongnam, Korea
charles@inje.ac.kr

Abstract. In NGN, many transport-related technologies were considered. One technology discussed is ATM (Asynchronous Transfer Mode) based MPLS (Multi-protocol Label Switching) for its provisioning QOS (Quality Of Service) commitment, traffic management aspects and smooth migration to the BcN in Korea. In this paper, we present preventive congestion control mechanism that uses network signaling information for detecting and fuzzy control to a SC(Slim Chance) of LSP (Labeled Switched Path) in ATM based MPLS system. Proposed control can handle 208% call processing and more than 147% success call, than those without control. It can handle 187% BHCA (Busy Hour Call Attempts) with 100 times less use of exchange memory.

Keywords: Congestion, MPLS, BcN, NGN.

1 Introduction

BcN is the Korean name of NGN that we resolutely called IP based B-ISDN. Most telecom operators use separate networks for voice and data services with different protocols and networking technologies. Current packet switched networks (IP routers) historically have never supported quality of service due to the fact that it only look at packets and do not keep any state information on individual flows. An individual flow might be a voice call, video call, file transfer, or web access. The need to integrate interactive and distribution services, as well as circuit and packet transfer modes into a universal broadband network are gradually increasing . For supporting QOS on the user side, network equipment venders try to introduce new mechanisms for converging circuit and packet switching into one. The nature of IP is random packet discarding during a congestion status. For provisioning QOS commitments in IP traffic, ATM based MPLS was introduced as a backbone system for traffic engineering and smooth migration from legacy networks. On February 28, 2005, there was a telecommunication disaster in Korea. The main reason for the disaster was call concentration at a specific time for telephone banking calls.

2 SC Definition , Detection and Control Mechanisms in PSTN

A call/ session for which the call/session completion ratio is lower than normal, is called a SC LSP. Such calls were hard to reach to the destination. A conventional

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 505 – 508, 2006.
© Springer-Verlag Berlin Heidelberg 2006

PSTN is directed to check the call completion ratio with regard to the SC candidate number that was entered by an operator.The originating exchange processes the call attempt, as an incomplete call by sending a busy tone to the user or to the switching system. However, when judging all of the busy tones to set an SC call should be because of numerous system errors. This is because the meaning of SC call is that the call completion ratios regarding the lack of resources in a specific toll/terminating exchange or that the user is busy. Therefore, it is necessary to measure the statistical probability of an incomplete call due to a lack of system resources and user busy. Two congestion control mechanisms were used in PSTN, that is "Automatic call gap" and "percentage based" congestion control. The time difference between congestion recognition and proper control action creates improper call/connection restrictions. The nominal time difference is 5 minutes for most conventional PSTN exchanges.

3 SC LSP Detection Method in BCN

This paper proposes SC LSP detection mechanisms using signaling information during the call active state. The ATM based MPLS system, since we do not have ATM aware terminals and protocol for a connection setup, most of cell/call level information that would be useful for knowing cell/call level detailed connection/ system status cannot be gathered. None of cell/call level information that are related the congestion can be available. New mechanisms should be introduced for controlling network congestion. Without the proposed mechanism, MPLS systems will easily face network congestion even with a small call/cell fluctuation. The followings are relevant information from the ATM based MPLS.

- EFCI/EBCI

EFCI/EBCI(Explicit Forward/Backward Congestion Indication) is a congestion notification mechanism which may be used to assist the network to avoid and recover from a congested state.

- ACL(Automatic Congestion Level)

ACL message is one of the information elements that can be generated by the call processing procedures. Among the RELEASE message, CAUSE information element that indicate detailed release causes is involved.

- AIS/RDI(Alarm Indication signal/Remote Defect indication)

AIS/RDI OAM cell will be generated by connection point or connection end point. VP/VC-AIS/RDI cells shall be generated and sent down stream on all affected active VPC (Virtual path Connection)/VCCs from the VP/VC Connecting point which detects the VPC/VCC defect at VP/VC level.

- Resource Management Cell

Resource management functions that have to operate on the same time-scale as the round trip propagation delays of an ATM connection may need ATM layer management procedures to use resource management cells associated with that ATM connection in ABR or ABT services.

Only AIS/RDI and EFCI/EFCI signal will indicates symptoms of congestion in the MPLS domain.

4 Fuzzy SC Control

Fuzzy logic was first proposed by Lotfi A. Zadeh of the University of California at Berkeley in a 1965 paper. He elaborated on his ideas in a 1973 paper that introduced the concept of "linguistic variables", which in this article equates to a variable defined as a fuzzy set. In telecommunication networks, several research studies which apply FCS to traffic control have been reported.One of previous research study using UCR(Uncomplete Call Ratio) with proper fuzzification rules are the same measurement interval as conventional PSTN exchanges, but they use UCR from current uncompleted rate at specific interval. However, gathering UCR data takes the same amount of time as PSTN control mechanisms. The main part of an FCS is a set of control rules of linguistic form that comprises an expert's knowledge. In this paper, we use NCS (Network Congestion Status) and CNCS (Change of NCS) as two inputs, to create a FCV (Fuzzy Control Value). These scaled inputs NCS and CNCS have their own fuzzy values in the set {NB (Positive Big), NS (Negative Small), ZO (Zero), PS (Positive Small), PB (Positive Big)}. The elements of the fuzzy set have their own triangular shaped membership functions. Using heuristic knowledge to decide the call blocking rate, we present the next three rules.

► If NCS is high, then it increases CNCS fast.

► If NCS is not high, but not too low neither, when CNCS is positive, it increases the call blocking rate fast, but if it is negative or equal to zero, it holds the current call blocking rate.

► If NCS is low, when CNCS is positive, it increases the call blocking rate moderately, and when CNCS is negative or equal to zero, it decreases call blocking rate slowly. Overall fuzzy control system in this paper is following

1. **Measurement:** $NCS\ , CNCS\ = NCSn\ - NCSn\ -1$
2. **Fuzzy value** $NCSF = NCS/\alpha, CNCSF = CNCS/\beta (Where 1 < \alpha, \beta < 100)$
3. **Apply Fuzzy rule(MS: Member Ship) (Sncs(f), SCNCS(f))**

$$Sncs\ (f) = \{(MS\ 1, MSvalue\ ), (MS\ 2, MSvalue\ )\}$$

4. **Apply Center of area**

$$CBR\ = \frac{\sum (Sncs\ (f) \times Scncs\ (f)(I,J)MSvalue\ ) \times (Sncs\ (f) \times Scncs\ (f)(I,J)Domain\ )}{\sum (Sncs\ (f) \bullet Scncs\ (f)(I,J) \bullet MSvalue\ )}$$

For example, if we use EFCI/EBCI and AIS/RDI, the composite values of NCS and CNCS are following.

$$NCS\ = \frac{NCSefci\ + NCSAIS}{2}$$

$$CNCS\ = \frac{CNCSefci\ + CNCSais}{2} = \frac{(NCSefcin\ - NCSefcin\ -1) + (NCsain\ \sin - NCSaisn\ -1)}{2}$$

Figure 2 explains the effect of call duration vs. admission calls. In multimedia service, actual call durations are important factor for determining MPLS system capacity.

Fuzzy NCS shows better performance than fuzzy UCR. However, beyond the engineering capacity quite different results are generated. We generated 0 to 40% EFCI/EBCI state randomly every specific intervals. In the Figure 3, we assumed that before the specific measurement interval (not shown in the figure), system was severe congestion status. The relevant previous UCR indicates 25% call blocking for any

reason. The figure3 shows relevant performance for each control methods. FCS control method that is based on current NCS and CNCS shows better performance. But in the x-axis near 86 times shows quite different behavior. It is because congestion status indicator (EFCI/EBCI) has abnormally generated on the time. We concludes UCS based control methods can not control a burst traffic, especially it can not control rapid change of call concentration and normal state.

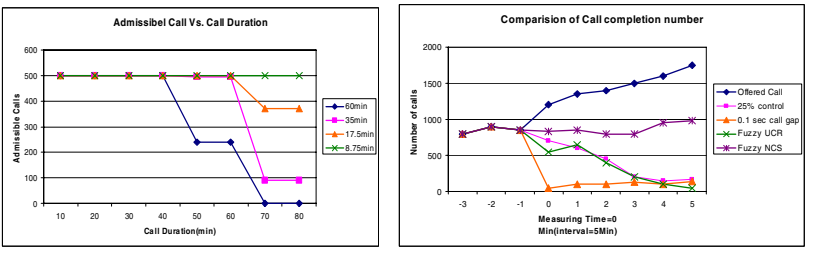


Fig. 2. Admissible calls vs. Call Duration and Fuzzy UCR vs NCS

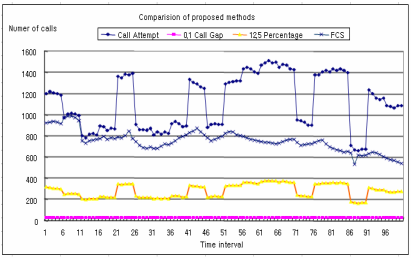


Fig. 3. Successful calls during rapid change of congestion status

5 Conclusions

In this paper, we present preventive congestion control mechanisms for detecting SC LSP in ATM based MPLS systems. In particular, we introduce a SC LSP detection method using network signaling information. SC LSP control can handle 208% call processing and more than 147% success call, than those without control. It can handle 187% BHCA with 100 times less use of exchange memory. We concluded that it showed fast congestion avoidance mechanism with a lower system load and maximized the efficiency of the network resources by restricting ineffective machine attempts.

Scalable DiffServ-over-MPLS Traffic Engineering with Per-flow Traffic Policing*

Djakhongir Siradjev, Ivan Gurin, and Young-Tak Kim**

Dept. of Information & Communication Engineering, Graduate School, Yeungnam University, 214-1, Dae-Dong, Gyeongsan-Si, Gyeongbook, 712-749, Korea
m0446086@chunma.yu.ac.kr, ivan_gurin@yumail.ac.kr, ytkim@yu.ac.kr

Abstract. This paper proposes a DiffServ-over-MPLS Traffic Engineering (TE) architecture and describes the implementation of its functional blocks on Intel IXP2400 Network Processor using Intel IXA SDK 4.1 framework. We propose fast and scalable 6-tuple range-match classifier, which allows traffic policing procedures to operate on per-flow level, and a scalable low-jitter Deficit Round Robin (DRR) scheduler that can provide bandwidth guarantees on LSP level. The proposed DiffServ-over-MPLS TE functional blocks have been implemented on Intel IXDP2400 platform for up to 4,096 flows mapped to L-LSPs, and can handle an aggregated traffic rate of 2.4Gbps.

Keywords: QoS, DiffServ-over-MPLS, scalability, network processors.

1 Introduction

The market for Internet Service Providers (ISP) is positioned to accept new premium service offerings such as voice over IP (VoIP), video on demand (VoD), and streaming television. QoS provisioning is essential for correct operation of abovementioned services.

Integrated services (IntServ) [1] and Differentiated Services (DiffServ) [2], standardized by IETF, do not meet current requirements to QoS provisioning. The former scales poorly and the latter can provide QoS only for traffic aggregates. Support of Differentiated Services [2] in MPLS [3] that was standardized by IETF can provide QoS guarantees, while keeping network resource utilization at high level, but also has no microflow policing. Also, Diffserv-over-MPLS [4] requires complex packet processing, especially on ingress nodes, and most of implementations separate DiffServ ingress node from MPLS ingress node. One of the performed tasks in DiffServ-over-MPLS processing is classification, which has either high memory complexity or high time complexity. Aggregated Bit Vector [5] is one of the algorithms that can fit requirements and limitations of network processor. Also DiffServ-over-MPLS TE requires scheduler to support high number of active queues,

* This research was supported by the MIC, under the ITRC support program supervised by the IITA.

** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 509 – 512, 2006.
© Springer-Verlag Berlin Heidelberg 2006

which can be changed dynamically with different bandwidth requirements, with several priority groups, and also have deterministic scheduling time.

This paper proposes a scalable and flexible DiffServ-over-MPLS TE with per-flow traffic policing. The design of processing pipeline and overall application is explained in detail. Per-flow policing on the edge and flow assignment to any created LSP allows fine-grained QoS provisioning. We implemented the proposed TE functions on Intel IXP2400 [6] network processor. Techniques and issues of range classifier implementation, based on ABV, are described, and low-jitter scheduling algorithm for Intel IXP 2400 network processor is proposed.

The rest of this paper is organized as follows. Section II describes the design and implementation of DiffServ-over-MPLS TE packet processing, and details of implemented functional modules. We cover the range classifier and low-jitter scheduler implementation plan in details, as one of the focuses of this paper. Section III analyzes current performance and theoretical final performance of the proposed architecture, and finally section IV concludes the paper.

2 DiffServ-over-MPLS TE on Network Processor

2.1 Diffserv-over-MPLS TE with Per-flow Policing Architecture

The proposed scheme is shown in the Fig. 1. All incoming IP packets are classified into separate flows. Each flow has its own traffic parameters that define Commited Information Rate (CIR) and Peak Information Rate (PIR) with corresponding burst

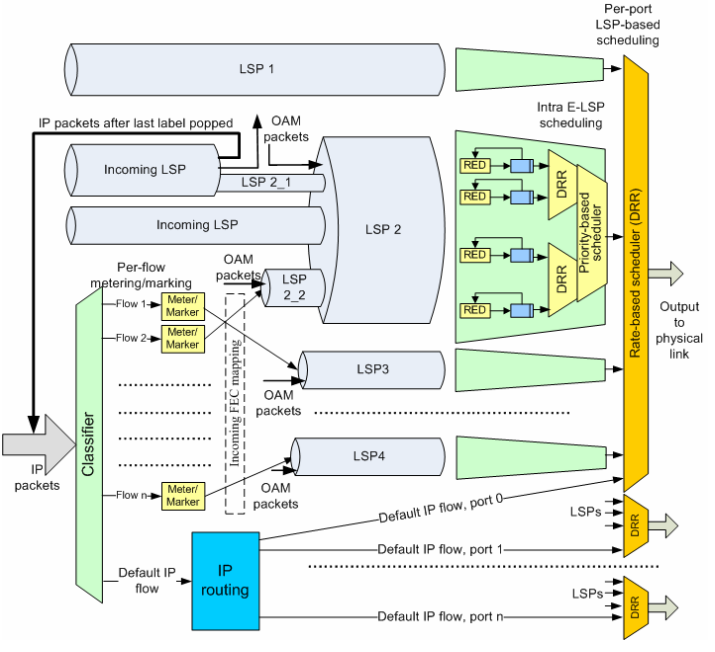


Fig. 1. Architecture of DiffServ-over-MPLS Traffic Engineering with Per-Flow Traffic Policing

sizes. After performing per-flow policing each flow is mapped to certain LSP. MPLS label stack operations are performed on incoming MPLS packets. As a result, a set of outgoing LSPs are produced. Each LSP has its assigned bandwidth, which is greater than or equal to the committed rate of all flows and LSPs that it carries. If there is exceeding traffic, currently unused bandwidth will be used to send it, or lower priority traffic will be preempted. The packets that could not be classified into flow will go through usual IP processing, and MPLS packets with last label popped, will be processed by classifier as usual incoming IP packet. In case of transit core router, processing is simpler, since no classification and flow metering is required. If some flow, requires very fine grained QoS, it can use dedicated LSP, although other flows of same class-type share the same LSP.

2.2 Scalable Range Classifier and Low-Jitter Scheduler

The packet classifier and scheduler, proposed in this paper, have IXA SDK compatible architecture and can be used in conjunction with other IXA SDK compatible modules. Both of them consist of 3 major parts: core component (CC) running on XScale processor, microblock running on microengines, and specific configuration utility running on XScale processor, which plays the role of interface between control plane and forwarding plane. Task of the core component is to configure the classifier operation and manage the rules database while microblock handles fast packet processing.

The original ABV proposal does not cover the method of search in each of fields describing only the aggregation technique that exploits the sparseness of bitvectors. In the proposed range classifier tries are used for search in certain dimension since tries can provide constant lookup time, at the expense of memory used. In order to decrease lookup time to some tolerable value, multibit tries, where each node represents several bits of prefix, are used. To increase update time for rules that have wildcarded fields, we introduce additional wildcard bitvector. Bitwise OR operation is performed on it and on the result of search, prior to combining results of individual search engines.

In order to achieve good scalability in packet scheduler constant computational complexity related to the number of queues in the system is essential. Bitvector-based schemes do not have such properties, due to linear complexity of multiplication, finding first set bit and other operations. In this paper, we propose a scalable low-jitter DRR scheduler with dual bidirectional linked list for constant complexity. The solution employed to gain the design objectives is storing the set of active queues in bidirectional linked list. Using two linked lists and prohibiting sending back-to-back packets, when there is more than one active queue, allows achieving another goal that is reducing delay jitter comparing to original DRR. Details of classifier and scheduler architecture are explained in [7].

3 Evaluations

IXDP2400 development platform was used for implementing proposed DiffServ-over-MPLS TE architecture. IXDP2400 platform has four 1Gigabit ethernet ports.

Smartbits 6000C traffic generator was used to generate network load. Tests were done for different types of processing. This system also has 4 1 Gigabit Ethernet ports. So this allows us to generate full load for network processor. Number of created rules is equal to 4,096 in all cases. First test was made with processing that receives IP packets, classifies them, and performs MPLS label pushing and related stack operation. According to the results, DiffServ-over-MPLS application with range classifier can support the line rate of OC-48 when the packet size is 512 bytes. Per-microblock processing latency measurement shows that the range classifier has the longest processing time. The detailed discussions of performance analysis and limitations of IXP2400 Network Processor are shown in [7]. For better performance, TCAM-based classifier should be used. Implemented scheduler supports 2^{18} queues and two priority groups.

4 Conclusion

In this paper, we proposed a scalable DiffServ-over-MPLS architecture for Network Processor with per-flow Policing. Also we proposed range classifier architecture for Network Processor, based on Aggregated Bit Vector scheme. Application shows tolerable results in performance test, and the performance can be easily improved by using TCAM-based classification, instead of software classification. Although the DiffServ-over-MPLS architecture is well-known, most of its implementations were designed for hardware, while the implementation on network processor can make it more flexible. The proposed DiffServ-over-MPLS TE implementation on network processor should help in the analysis of functional blocks and improving them in future. Future works include performance enhancement and development of functional modules of control plane.

References

1. R. Braden et. al., “Integrated Services in the Internet Architecture: an Overview,” RFC 1633, IETF, June 1994.
2. S. Blake et al., “An Architecture of Differentiated Services,” RFC 2475, IETF, December 1998.
3. E. Rosen et al., “Multiprotocol Label Switching Architecture,” RFC 3031, IETF, January 2001.
4. F. Le Faucheur, editor, “Multiprotocol Label Switching (MPLS) support of Differentiated Services,” RFC 3270, IETF, April 2002.
5. Florin Baboescu and George Varghese, “Scalable Packet Classification,” IEEE/ACM transactions on networking, vol. 13, No. 1, February 2005.
6. M. Shreedhar, G. Varghese, “Efficient Fair Queuing Using Deficit Round-Robin,” IEEE/ACM Transactions on Networking, Vol.4, No.3, June 1996.
7. Djakhongir Siradjev, Ivan Gurin, Seung-Hun Yoon and Jeong-Ki Park, “DiffServ-over-MPLS TE on IXDP2400,” YUANTL-TR-06-NP-DSMPLS-01, July, 2006.

On the Dynamic Management of Information in Ubiquitous Systems Using Evolvable Software Components*

Syed Shariyar Murtaza[1], Bilal Ahmed [2], and Choong Seon Hong[3,**]

Dept. Of Computer Engineering Kyung Hee University, South Korea
shariyar@networking.khu.ac.kr, bilal@oslab.khu.ac.kr,
cshong@khu.ac.kr

Abstract. Rapid evolution in ubiquitous systems is complicating its management. In the realm of ubiquitous computing, the domain specific ontologies are constructed manually by experts, which is quite a toiling task. We proposed to extract ontologies dynamically for ubiquitous systems. We also argue that the components of the software middleware are the specialized research areas with different subject orientation e.g. context reasoning, service discovery etc[1] and following an evolvable component oriented approach would let the components of the software evolve independently of other software components, while making them interoperable irrespective of versions and vendors.

1 Introduction

With the ubiquity of emerging ubiquitous devices, our access to data would evolve exponentially and this continuous evolution in the information and interfaces would overwhelm the humans. Therefore, we need a mechanism which could evolve both in terms of information and software.

Different types of software infrastructures for the ubiquitous systems have been developed in the past years like Gaia [5], Solar System [8] and Context Toolkit [9] etc. But, they didn't consider the gradual evolution of the ubiquitous environments and the composition of ontologies is also done manually. To cope with evolution and interoperability, it is necessary to separate the overall environment into smaller logical groups or modules. Also, in these rapidly evolving ubiquitous environments, it may not even be desirable to build a comprehensive ontology, because the interfaces of appliances or devices are liable to change. Therefore, we seek to develop onotologies dynamically. Number of researchers has proposed different dynamic ontology extraction algorithms for text documents and web pages e.g. Chung Hee Hwang [2], Giovanni et al [3], and Zhan Cui et al. [4] etc. But, the dynamic extraction of ontologies in the ubiquitous environment to learn the information about the environment and user to provide him/her the seamless effect in the interaction with devices and services around him was never considered.

* This wok was supported by MIC and ITRC Project.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 513 – 516, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 System Description

Our system contains two major applications: Server application and User application, along with several device applications for each home appliance (TV, Bulb etc).

The system employs UPnP [7] protocol and is developed using C# and Intel's UPnP SDK. A self explanatory description of the system is shown in figure 1.

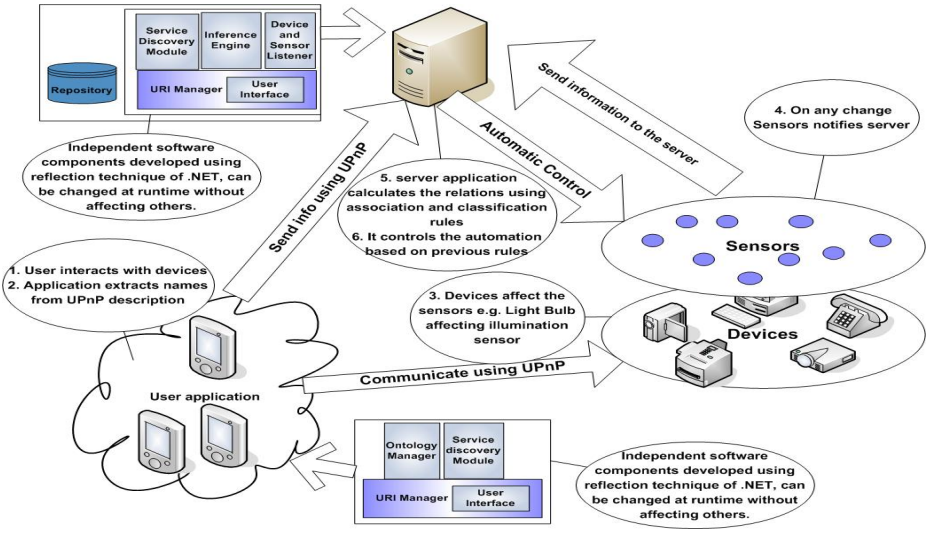


Fig. 1. System Overview

On user's interaction with devices, the user application extracts device name from the UPnP based device description by extracting the stop words from the manufacturer tag, then eliminating stop words[1] from the common string of the "Model Name" and "Model Description" tags of UPnP device description. This is shown in figure 2. The next step is to find the relations between devices. This is done at the server by finding the device and sensor relation using association rules [6]. The general template of the association rule is given below, where D manifests device and S manifests sensor. These association rules then become the basis of the device and schedule classification rules.

$$\mathbf{D_1 \cap D_2 \cap D_3 \cap \ldots . D_n \rightarrow S_i} \quad (1)$$

Device Classification Rules:

Let X={X1, X2….. Xn} be the devices and Xi, Xj $\in$ X where i,j=1 to n

1) If X1, X2 devices affect some common sensors and other uncommon sensors then they can be regarded as similar devices.
2) Devices always affecting same common sensors are same devices.
3) Same Device: If X set of devices is the 'Same Device' and if Xi has the request of use and it is not available then use Xj such that Xj value of effect

[1] Here, stop words represent those words which are not the part of actual name.

on the sensors (e.g. illumination intensity level) is less than or equal to Xi greater than all the value of effects in Y=X-Xv : Xv={Xi,Xj}

4) Environment Control: If X set of devices is similar and the request or priority of use of the value of effect from the user is c (like Noise level 30 of room) then sum of value of effect of all the devices in X should be less than equal to c.

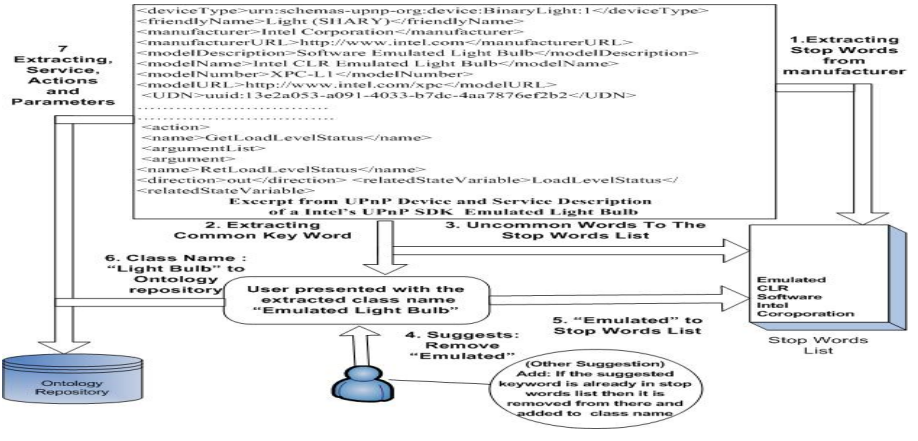


Fig. 2. Process of Extracting Device Level Concepts

Schedule Classification Rules:

1) Divide the time duration of a day into different slices of time.
2) For every slice of time calculate the highly used devices (devices having sup port greater than 50) and start them automatically.
3) Similarly, calculate low used devices (devices having sup port from 25 to 50) and leave them in their current state, while turn off all other devices
4) If the device throws error or not found then go for same device rule (1).

3 Evaluation

We considered a single room, a user and designed simple UPnP home appliances e.g. dimmable light bulb, medium size microwave oven, air conditioner with temperature control facility and etc. Similarly, we deployed different sensors like illumination sensor, temperature sensor and noise sensor. We performed the experiment for a week and divided the time durations into different slices according to the schedule rule for this experiment: Breakfast and wakeup timing, office timing, dinner and relaxation timings and sleep timing. Initially, the user application extracted the names then the server application calculated the relationships according to the association and classification rules of the previous section. We have shown these ontologies and their results in the form of a tree in the figure 3.These results allow the system to automate the environment and provide a seamless effect in a number of ways e.g. When the user gets up in the morning then according to the schedule rules of the previous sections the system will automatically turn on the Light bulb, but if it is not available then by same device rule it will turn on the lamp. Similarly, many other scenarios are possible.

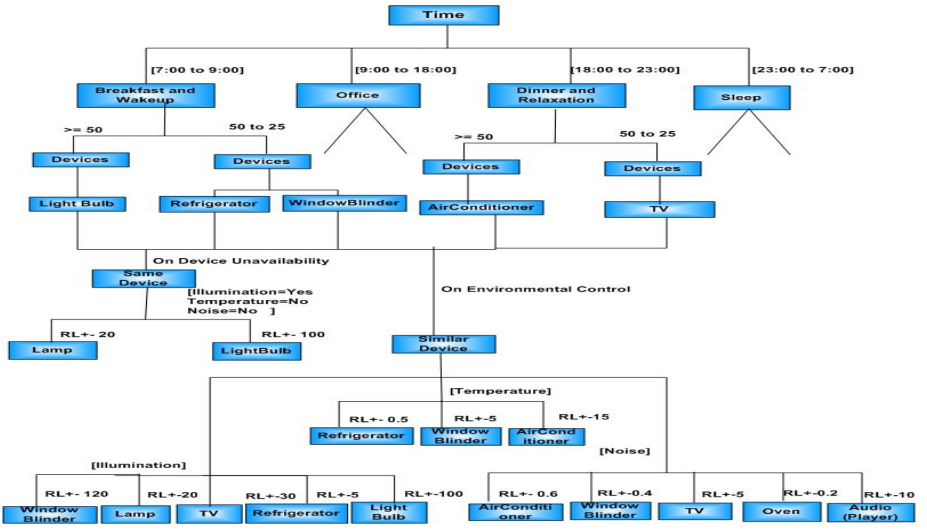


Fig. 2. Tree Diagram for Classification Relationship

In the future, we would like to publish these extracted ontologies in the form of RDF to reuse and share this work with other systems

References

1. Syed Shariyar Murtaza,Choong Seon Hong," An Evolvable Software Architec-ture for Managing Ubiquitous Systems",Proc. of 8th Asia Pacific Network Operation and Managment Symposium (APNOMS), September 2005,pp 400-409
2. Chung Hee Hwang, "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information", proc. of the 6th International Workshop on Knowledge Representation meets Databases, July ,1999.
3. Giovanni Modica, Avigdor Gal, and Hasan M. Jamil, "The Use of Machine-Generated Ontologies in Dynamic Information Seeking", Proc. of the 9th In ternational Conference on Cooperative Information Systems, 2001, 443- 448
4. Zhan Cui, Ernesto Damiani, Marcello Leida, Marco Viviani, "OntoExtractor A Fuzzy-Based Approach in Clustering Semi-structured Data Sources and Meta data Generation", proc. of Knowledge-Based Intelligent Information and Engineering Systems, LNCS, 05
5. A Ranganathan, Roy Campbell: A Middleware for Context -Aware Agents in Ubiquitous Computing Environments, proc. of ACM/IFIP/USENIX Middleware Conference,June,03
6. Han, Jiawei. Kamber, Micheline.: Data Mining: Concepts and Techniques, Morgan Kauf mann Publishers (2001): ISBN: 1558604898.
7. UPnP forum, http://ww.upnp.org
8. Guanling Chen and David Kotz.: Solar: An Open Platform for Context –Aware Mobile Applications, proc of the First International Conference on Pervasive Computing (Pervasive 2002), Switzerland, June, 2002.
9. Context Toolkit project http://www.cs.berkeley.edu/~dey/context.html

A Shared-Memory Packet Buffer Management in a Network Interface Card

Amit Uppal and Yul Chu

Electrical and Computer Engineering Department
Mississippi State University
P.O. Box 9571
Mississippi State, MS 39762, USA
{au6, chu}@ece.msstate.edu

Abstract. This paper proposes a dynamic shared-memory packet buffer management algorithm for a protocol processor in a network terminal. The protocol processor is located in a network interface card (NIC). In general, two types of packet buffer management algorithms, static and dynamic, can be used in a NIC; the dynamic buffer management algorithms work better than the static ones for reducing the packet loss ratio. However, conventional dynamic buffer management algorithms do not provide even packet losses to all the applications. Therefore, we propose an algorithm to enhance even packet losses and the proposed algorithm improves the packet loss ratio by 11.80% to 16.70% compared to other conventional dynamic algorithms.

1 Introduction

A shared-memory packet buffer in a network interface card (NIC) is a large shared dual-ported memory [4]. Packets for each application are multiplexed into a single stream. In an intelligent NIC, the packet buffer management algorithm determines whether to accept or reject each packet. The accepted packet is placed into a logical FIFO queue; each application has its own queue in a packet buffer [1]. The accepted packet remains in a buffer until the application retrieves it from the buffer. These accumulated packets in the buffer can reduce the available buffer space for a next incoming packet. Once the buffer is full, further incoming packets will be dropped. Therefore, it is important to reduce packet loss ratio to support any end-to-end application in a computer network [4]. Buffer management algorithms in a NIC determine how the buffer space is distributed among different applications. The design of a buffer management algorithm needs to consider the following two factors [1]: 1) Packet loss ratio and 2) Hardware complexity. We propose an efficient buffer management algorithm called Evenly Based Dynamic algorithm (EBDA); EBDA provides fairness to all the applications while reducing the packet loss ratio.

Tomas Henriksson, et al. [1] proposed protocol processor architecture to offload the host processor for a high-speed network. The new packet reception, move the layer 3 and layer 4 processing to an intelligent NIC. The main goal of the protocol processor is to handle the TCP/IP or the UDP/IP processing at a wire speed.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 517 – 520, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Buffer Management Algorithms in a NIC

We will discuss two types of buffer management algorithms in a NIC: 1) Static Threshold Scheme algorithms: Completely Partitioned (CP), and Completely Shared (CS); 2) Dynamic Threshold Scheme algorithms: Dynamic Algorithm (DA), and Dynamic Algorithm with Dynamic Threshold (DADT).

Kamoun and Kleinrock [4] proposed CP. In CP, the total buffer space '*M*' is equally divided among all the applications (N). Packet loss for any application occurs when the buffer space allocated to that application becomes full.

In CS [4], packets are accepted as long as there is some space left in a buffer, independent of the application to which a packet is directed. This algorithm utilizes the whole buffer space. Packet loss occurs only when the buffer is full.

In DA, packets for any application are accepted as long as the queue length for the application is less than its threshold value. Packet loss occurs only when the queue length of an application exceeds its threshold value. At any instant 't', let T (t) be the control threshold and let $Q_i(t)$ be the length of queue 'i'. If Q (t) is the sum of all the queue lengths [5], and '*M*' is the total buffer space, the controlling threshold will be

$$T(t)=\alpha*(M-Q(t)) \quad (1)$$

where α is some constant. This algorithm is robust to changing load conditions in traffic, and it is also easy to implement in hardware.

The DADT [6] works like DA. In this algorithm, the alpha (α) value is different for different applications and is dependent on the packet size of an application. Unlike DA, different applications do not have the same threshold value. By varying the threshold value, DADT does not allow queues with the largest packet size to fill the buffer at a faster rate. In DADT, we have

$$T_i(t)=\alpha_i*(M-Q(t)) \quad (2)$$

where α_i is the proportionality constant and vary for each application.

3 EBDA

DADT reduces the overall packet loss ratio by giving less threshold value to the applications with larger packet sizes. This results in an increase in packet losses for applications with larger packet sizes, thus resulting in reducing fairness for applications with large packet sizes. Therefore, we proposed EBDA that will take fairness among applications and packet sizes of applications into consideration while allocating buffer space to each application.

Fig. 1 shows the flowchart of EBDA**.** In EBDA, the threshold value for an application '*i*' with packet size '*psize(i)*' less than the average packet size *(Σpsize(i)/n)* is calculated as shown in equation 3; where n is the number of total applications. For an application with a packet size greater than the average packet size, the threshold value is calculated as shown in equation 4.

Our simulation results have shown that by taking packet size factor in the summation as in equation 3 and 4, instead of multiplication for determining the

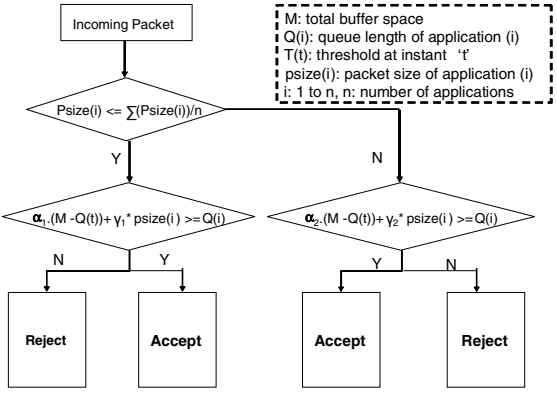


Fig. 1. Flowchart of EBDA

threshold value for the application, we can reduce the overall packet loss ratio as well as distribute the packet losses more evenly among the different applications.

$$T(t)=\alpha_1 * (M-Q(t))+ \gamma_1 * psize(i) \quad (3)$$

$$T(t)=\alpha_2 * (M-Q(t))+ \gamma_2 * psize(i) \quad (4)$$

The optimum alpha1 (α_1), alpha2 (α_2), gamma1 (γ_1), and gamma2 (γ_2) values as shown in equation 3, 4 are determined through simulations.

4 Simulation Results and Analysis

We implemented a traffic mix with the average network traffic loads according to [2]. We have taken six applications for our simulations with packet sizes as 8,2,8,1,4,16 Bytes respectively. We have taken a buffer size of 600 packets.

For DA, optimum value of alpha for minimum packet loss ratio for average traffic load comes out to be 16 and for DADT, optimum value of alpha for different applications comes out to be 16,14,16,14,16,8 respectively. For EBDA, optimum values of alpha1, gamma1, alpha2, and gamma2 come out to be 16, 4, 64, and 64 respectively.

Fig. 2 shows the performance of the three algorithms (EBDA, DA, and DADT) for different loads. Load has been varied from 0.5 to 0.9. As seen in Fig. 2, EBDA has the least packet loss ratio for all of loads. Notice that the performance difference increases more at higher loads. As the load is increased, applications with larger packet size tend to increase their queue length to values greater than their threshold values frequently. Since, EBDA utilizes the buffer space more efficiently, providing fairness to all the applications; EBDA can reduce the packet loss ratio significantly.

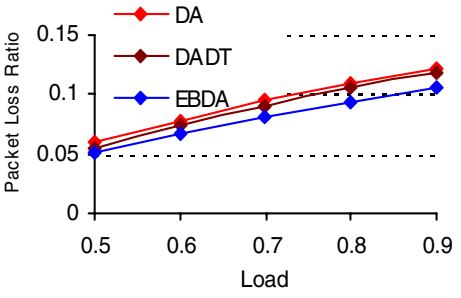


Fig. 2. Packet loss ratio vs. Load for EBDA, DADT, DA for the average traffic load

5 Conclusions

Three buffer management algorithms are implemented for our simulations: 1) Dynamic algorithm (DA); 2) Dynamic Algorithm with Dynamic Threshold (DADT); and 3) Evenly Based Dynamic algorithm (EBDA)**.** EBDA provides more fairness to all the applications and utilizes buffer space efficiently, which makes it different from DA and DADT. For the traffic mix with the average network traffic loads, the EBDA improves the packet loss ratio by 16.7% as compared with DA, and by 11.8% as compared with DADT.

References

1. T. Henriksson, U. Nordqvist, D. Liu, Embedded Protocol Processor for fast and efficient packet reception, IEEE Proceedings on Computer Design: VLSI in Computers and Processors, vol. 2, pp. 414-419, September 2002.
2. U. Nordqvist, D. Liu, Power optimized packet buffering in a protocol processor, Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, vol. 3, pp. 1026-1029, December 2003.
3. M. Arpaci, J.A. Copeland, Buffer Management for Shared Memory ATM Switches, IEEE Communication Surveys, First Quarter 2000.
4. F. Kamoun, L. Kleinrock, Analysis of Shared Finite Storage in a Computer Network Node Environment under General Traffic Conditions, IEEE Transactions on Communications, vol., COM-28, pp. 992-1003, July 1980.
5. A. K. Choudhury, E.L. Hahne, Dynamic Queue Length Thresholds for Shared-Memory Packet Switches, IEEE/ACM Transactions on Communications, vol. 6, no. 2, pp. 130-140, April 1998.
6. Yul Chu, Vinod Rajan, An Enhanced Dynamic Packet Buffer Management, In the proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05), Cartagena, Spain, June 2005.

An Adaptive Online Network Management Algorithm for QoS Sensitive Multimedia Services

Sungwook Kim[1] and Sungchun Kim[2]

[1] Department of Computer Science, Sogang University,
Shinsu-dong 1, Mapo-ku, Seoul, 121-742, South Korea
swkim01@sogang.ac.kr
[2] Department of Computer Science, Sogang University,
Shinsu-dong 1, Mapo-ku, Seoul, 121-742, South Korea
ksc@mail.sogang.ac.kr

Abstract. The explosive growth of new multimedia services over the Internet necessitates efficient network management. Improved network management systems are expected to simultaneously provide diverse multimedia traffic services and enhance network performance. In this paper, we propose a new online network management algorithm that implements adaptation, reservation, and call-admission strategies. Our online approach to network management exhibits dynamic adaptability, flexibility, and responsiveness to the current traffic conditions in multimedia networks.

1 Introduction

Multimedia is a keyword in the evolving information age of the 21st century. In recent years, the growth of multimedia applications that can be represented through audio and, video streams, images and animations has greatly increased the research interest in quality of service (QoS). The ultimate goal of network QoS support is to provide applications with a high-quality data delivery service [1]-[2].

The different types of multimedia service provided over networks not only require different amounts of bandwidth but also have different policy assumptions. The heterogeneous multimedia data usually categorized into two classes according to the required QoS: class I (real-time) and class II (not real-time). The class I data type has a higher priority than class II data type and so a multimedia network should take into account the prioritization among different multimedia traffic services [1]-[2].

Internet multimedia services have continued to emerge thanks to the benefits associated with the stateless architecture of the Internet Protocol (IP), and has made the provision of QoS-sensitive multimedia data services an area of great importance. However, the Internet is currently not designed to support the performance guarantees such as bounded delay and minimum throughput that are generally required for the higher priority class I applications [3].

QoS provisioning requires the management of admission control [1]-[3]. An essential role of call admission is to protect admission-controlled traffic from non-admission-controlled traffic. The bandwidth of a link on the Internet is shared

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 521 – 524, 2006.
© Springer-Verlag Berlin Heidelberg 2006

dynamically between class I and class II data services. Since - each service has different operational requirements - different admission control rules are applied to each application type. For example, based on traffic priority, there can be class I applications with strict admission control rules and class II applications with non-controlled admission rules [3].

The call-admission mechanism is used to reserve bandwidth. In a reservation procedure, some of the available bandwidth is reserved for use by higher priority traffic services. Therefore, admission-controlled class I data services can actually benefit from bandwidth reservations for QoS guarantees [1]-[3].

Efficient network management requirements control decisions that are dynamically adjustable. However, at any point in time the future rate of traffic arrival is generally not known, and there can be dramatic short-term variations in traffic patterns. These control decisions therefore have to be made in real time.

Online algorithms [4] are natural candidates for the design of efficient control schemes in QoS-sensitive multimedia networks. An algorithm employing online computations is called an online algorithm, and the term 'online computation problem' refers to decision problems where decisions must be made in real time based on past events without information about the future.

Motivated by the above discussion, we propose a new online network management algorithm for QoS-sensitive multimedia networks. Our algorithm is designed to handle control decisions in an online manner. Due to the uncertain network environment, an online strategy based on real time measurements of the current network conditions plays an important role in determining the network performance.

The important features of our algorithm are (i) the inclusion of a QoS guarantee that does not reduce the network capacity, (ii) the ability to adaptively control congestion so as to maximize network performance, (iii) a low complexity that makes it practical for real-world implementation, and (iv) the ability to respond to current network traffic conditions for appropriately balancing the performance between contradictory QoS requirements.

2 Proposed Network Management Algorithm

There are many alternative ways to implement admission control in IP networks with a bandwidth broker (BB) being popular in recent years [3].

Call admission decisions are made according to bandwidth reservation. Therefore, this paper focuses on the BB approach for class I call-admission and reservation controls. In each routing domain, a BB controls bandwidth reservation and makes class I call-admission decisions based on the measured link load information on source-to-destination paths. Other BBs in neighboring domains may also have to be consulted if the destination is not in the same domain. If there is sufficient available bandwidth on the path, a class I connection request is admitted and - the requested bandwidth is allocated for all links along the path; - otherwise, this type of call request is rejected.

In multimedia networks, class I traffic services create very regular traffic patterns from source to destination, and service levels are specified as a desired peak bit-rate for a specific flow. Therefore, controlling call admission and bandwidth reservation will allow the QoS for higher priority class I data services to be guaranteed. For

adaptive online control, our proposed mechanism provides a coordination paradigm by employing two approaches: (i) an event-driven approach for call admission and (ii) a time-driven approach for bandwidth reservation. This cooperative combination provides adaptive online QoS management in multimedia networks.

The purpose of bandwidth reservation is to ensure some network capacity for class I traffic. However, using a strict reservation policy can result in a serious underutilization of the bandwidth efficiency. Therefore, reservation strategy allows trading-off between bandwidth utilization and QoS guarantees.

For a given traffic load, there is an optimal amount of bandwidth that should be reserved, but this amount naturally varies with the network traffic. To determine the optimal amount dynamically, we partition the time-axis into equal intervals of length *unit_time*. Our proposed online algorithm adjusts the amount of reserved bandwidth (Res_B) based on real time measurements during every *unit_time*.

To maintain the reserved bandwidth close to the optimal value, we define a traffic window, that is used to keep the history of class I traffic requests (W_{class_I}). The traffic window is of size [t_c - t_{class_I}, t_c], where t_c is the current time and t_{class_I} is the window length, and this size can be adjusted in time steps equal to *unit_time*. The traffic window size is increased and decreased if the call blocking probability (CBP) for new class I service requests is larger or smaller than its predefined target probability (P_{class_I}), respectively. The values of Res_B can be estimated as the sum of requested bandwidths by class I calls during the traffic window:

$$Res_B = \sum_{i \in W_{class_I}} (B_i \times N_i) \tag{1}$$

where N_i and B_i are the number of class I data requests and the corresponding bandwidths of data type i, respectively. Therefore, by using this traffic window, we can adjust the amount of the reserved bandwidth (Res_B) at every *unit_time*, which is more responsive to changes in the network condition after the bandwidth has been reserved.

A BB monitors the traffic load of each link in its own domain and updates the link database by calculating the current amount of unoccupied (UB_{link}) and reserved (RB_{link}) bandwidth. The total available bandwidth (AB_{link}) for class I services can be estimated as the sum of UB_{link} and RB_{link}:

$$\mathrm{AB_{link}} = RB_{link} + UB_{link} \tag{2}$$

Based on $\mathrm{AB_{link}}$ information, our call-admission mechanism can measure the minimum available link bandwidth from source node i to destination node j ($MAB_{path(i,j)}$) according to

$$\mathrm{AB_{path(i,j)}} = \min_{link \in path\,(i,j)} (AB_{link}) \tag{3}$$

$MAB_{path(i,j)}$ is used to makes call-admission decisions.

The bandwidth utilization will be suboptimal - if class I services claim all the reserved bandwidth. Since link bandwidth should be shared dynamically between class I and class II traffic, our mechanism allows reserved bandwidth to be borrowed temporarily for under-allocated or burst periods of existing class II call connections.

This adaptive bandwidth allocation of reserved bandwidth reduces bandwidth waste associated with the reservation process.

3 Summary and Conclusions

Heterogeneous multimedia networks such as the Internet should provide QoS guarantees through scalable service differentiation, especially under conditions of heavy network congestion. In this paper, we propose the network management algorithm that balances between a high bandwidth utilization and QoS provisioning under a wide variety of network traffic loads. This is achieved by applying ideas from the online call-admission mechanism to network congestion controls, with this collaborative combination supporting QoS provisioning without sacrificing bandwidth utilization.

References

1. Sungwook Kim and Pramod K. Varshney, “An Adaptive Bandwidth Allocation Algorithm for QoS guaranteed Multimedia Networks ”, *Computer Communications 28*, pp.1959-1969, October, 2005.
2. Sungwook Kim and Pramod K. Varshney, “An Integrated Adaptive Bandwidth Management Framework for QoS sensitive Multimedia Cellular Networks”, *IEEE Transaction on Vehicular Technology*, pp.835- 846, May, 2004.
3. J. Lakkakorpi, O. Strandberg and J. Salonen, "Adaptive Connection Admission Control for Differentiated Services Access Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 10, October 2005, pp. 1963-1972.
4. Yossi Azar, *Online Algorithms - The State of the Art*, Springer, 1998.

Improved Handoff Performance Based on Pre-binding Update in HMIPv6

Jongpil Jeong, Min Young Chung, and Hyunseung Choo*

Intelligent HCI Convergence Research Center
Sungkyunkwan University
440-746, Suwon, Korea
+82-31-290-7145
{jpjeong, mychung, choo}@ece.skku.ac.kr

Abstract. In this paper, an efficient neighbor AR (Access Router) discovery scheme and handoff procedure using neighbor information are proposed. It allows each AR and Mobility Anchor Point (MAP) to know neighboring ARs and MAPs, and therefore, a mobile node (MN) can perform the handoff process in advance, using the proposed handoff mechanism. It is important to note that the Inter-MAP domain handoff improvement of the proposed scheme is up to about 57% and 33% for handoff latency in comparison of the Hierarchical Mobile IPv6 (HMIPv6) and the Hierarchical Mobile IPv6 with Fast handover (F-HMIPv6), respectively. The proposed scheme is approximately two times better than the existing one in terms of the total signaling costs through performance analysis. Therefore, it is sufficiently satisfied with the requirements of real-time applications, and seamless communication is expected.

1 Introduction

Mobile IPv6 [1,2] handoff incurs high handoff latency, data loss, and global signaling. Basically, Fast Mobile IPv6 (FMIPv6) [3] reduces handoff latency by link layer (L2) triggers and prevents data loss by creating a bi-directional tunnel between a mobile node's previous subnet's access router (oAR) and next subnet's access router (nAR). HMIPv6 [4] prevents global handoff signaling by appointing a MAP that acts like a local Home Agent (HA). In Mobile IPv6 and HMIPv6, no information is exchanged among ARs. Therefore, only after completing L2 handoff, an MN can receive information regarding the AR, to which the MN will handoff via an agent advertisement message. On-going communication sessions with other hosts are impossible before completion of this handoff process, which is the major portion of overall handoff latency [5,6].

In the proposed mechanism, an AR can learn information regarding its geographically adjacent ARs - typically, global address, L2 identifier, and the prefix information of ARs that are currently being advertised. The current AR that the MN is visiting would be able to inform the MN of the prefix information of ARs to which the MN would likely handoff. After completion of the Address Autoconfiguration (AA) process, the MN transmits an incomplete binding update

* Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 525–529, 2006.
© Springer-Verlag Berlin Heidelberg 2006

message to a MAP, and then the MAP performs a Duplicate Address Detection (DAD) process using this message. Through this signaling flow, there is a remarkable decrease in handoff latency.

2 Related Works

This section provides a brief overview of the differences to be taken into account for the various approaches to reduce the handoff latency. *Basic MIPv6* [1,2], *Anticipated FMIPv6* [3], *Hierarchical Mobile IPv6* [4], *Hierarchical Mobile IPv6 with Fast handover (F-HMIPv6)* [3,8], *FMIPv6 for HMIPv6 (FF-HMIPv6)* [9].

3 The Proposed Protocol

In Mobile IPv6, after an MN moves from one subnet to another and performs the AA process, the MN informs the current network of its global address. The AR receiving this message verifies whether the MN's address can be used by the DAD. The ARs and MAPs know their respective neighboring AR's address information using the neighbor discovery scheme. When the MN senses that it will perform handoff, it transmits a ***handoff solicitation message*** to the current AR, and the AR transmits an ***advertisement message*** with options. When the MN receives this message, it performs an AA process to the new Care-of-Address (CoA) in advance before the actual handoff, and transmits a ***Pre-BU message*** to the AR, to which the MN will handoff. The ***Pre-BU message*** is the same as the ***typical BU message*** with an additional reserved field of the mobility header. The AR (AR and MAP in case of Inter-MAP domain handoff) performs the DAD and records the address as an incomplete state of the MN's address, meaning that it will not be used for routing of the MN until the MN transmits the ***real BU message*** after the actual handoff. After L2 handoff, the MN transmits the ***BU message*** to the new MAP via the new AR. The new AR and MAP finish the handoff process and directly progress routing by changing the MN's address in an incomplete state into a complete one, without the DAD process.

When the MN enters another regional network, it sends the BU to the first AR of the subnet. The AR relays it to the MAP2 (new MAP), and the MAP2 sends it back to the MAP1 (old MAP). When the MAP2 receives its message, it compares the one to the MAP list and finds the MN's field. And it updates the current MAP address of the MN. The MAP1 already has a table to manage the neighboring MAPs and generates another table to manage the MN. In addition, MAP1 registers all Correspondent Nodes (CN) connected to HA and MN using its own Regional CoA (RCoA). An MN sends the MAP1's RCoA and initial MAP's RCoA to the MAP2 when MN registers to the MAP2. The MAP2 sends its RCoA and initial MAP's RCoA to the MAP1, and then the MAP1 compares the initial MAP's RCoA with itself. Therefore, when HA transmits data, the data is delivered by this route. This scheme saves signaling costs in comparison with the existing schemes, like HMIPv6. Similarly, when a MN moves from MAP2 to MAP3, the MN transmits the registration message (MAP2's RCoA, initial

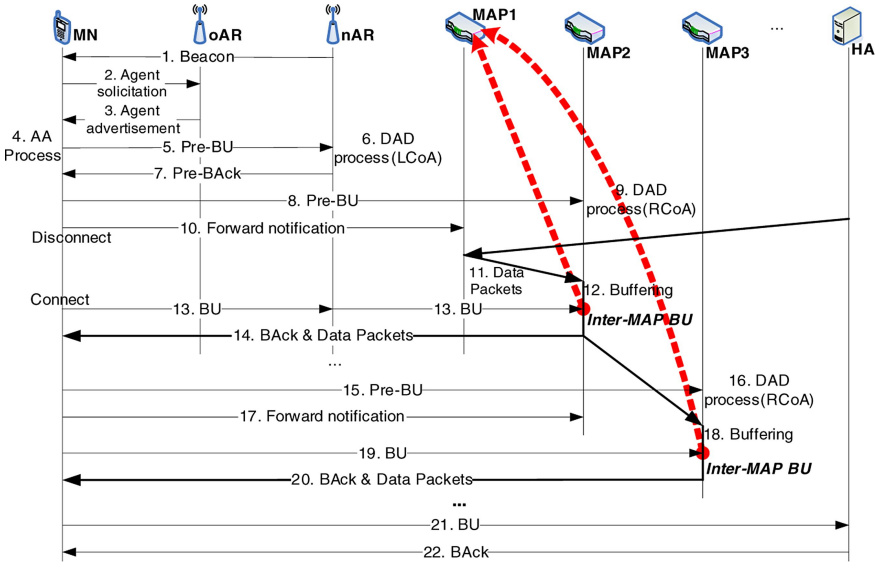


Fig. 1. Signal flows of Inter-MAP domain handoff

registration MAP's RCoA) to MAP3, and then MAP3 transmits two messages to MAP1. In this case, MAP1 deletes MAP2's RCoA in the MN's list since it contains the MAP's RCoA and records MAP3's RCoA. MAP1 and MAP3 are then linked through the registration. Therefore, the proposed scheme is not required to send the binding information to the HA and CNs. Fig. 1 represent the signal flows of the proposed handoff process, and only the Inter-MAP domain handoff is described in detail.

4 Performance Evaluation

First, the handoff latency is studied - the sum of the L2 and L3 handoff latency. The handoff latency of basic Mobile IP is defined as the period between the disconnection of the MN's wireless link and reception of AR's binding acknowledgement by MN. It is used for Intra-MAP domain handoff. In Inter-MAP domain handoff, handoff latency is the time from when MN triggers link-down in the current network to when the MN receives HA's first binding acknowledgement after handoff. The proposed scheme presents the minimum latency. For Inter-MAP domain handoff, HMIPv6 presents the largest handoff latency. Note that the Inter-MAP domain handoff improvement of the proposed scheme is very large. This is due to the fact that the proposed scheme performs the AA and DAD processes in advance. MIPv6 presents many packet losses due to the large handoff latency, and HMIPv6 presents decreased packet losses than the base MIPv6 through the advantages of the MAP entity. The proposed scheme presents superior performance without packet loss. In the proposed scheme, the

MAPs buffer and forward packets during the handoff period, this improves overall performance.

From analysis [7], the total signaling cost of the packet forwarding scheme is smaller than the HMIPv6 up until 8 forwarding steps ($q \leq 8$). q represents the maximum number of forwarding link. However when $q \geq 9$, the cost of the packet forwarding scheme becomes greater than the HMIPv6 one, and the MN sends the registration message to the HA and the CNs, removing all of the previous links among MAPs. Although at $q \geq 9$, the total signaling cost changes little. In the worst case, the total signaling cost of the proposed scheme is smaller than those of the HMIPv6 and the Forwarding up until 17 forwarding steps ($q \leq 17$).

5 Conclusion

An efficient neighbor AR discovery scheme and the handoff procedure using the neighbor information are proposed. The proposed neighbor ARs discovery scheme allows each AR and MAP to know its neighboring ARs and MAPs, and therefore the MN can perform the handoff process in advance. According to the simulation study, the proposed handoff mechanism demonstrates improved performance over existing mechanisms, due to the fact that the proposed scheme performs AA and DAD processes in advance. The proposed scheme does not transmit the BU to the CNs and the HA when the MN moves among adjacent MAPs. Instead, the current location of the MN is informed by transferring the modified BU to the previous MAP. According to the results of the performance analysis, the proposed scheme is approximately two times better than the HMIPv6 one in terms of the total signaling costs through performance analysis.

Acknowledgment

This research was supported by Ministry of Information and Communication, Korea under ITRC IITA-2005-(C1090-0501-0019) and grant No. R01-2006-000-10402-0 from the Basic Research Program Korea Science and Engineering Foundation of Ministry of Science & Technology.

References

1. T. Narten *et al.*, "Neighbor Discovery for IPv6," Internet-Draft, October 2005.
2. D. Johnson *et al.*, "Mobility Support in IPv6," IETF RFC 3775, June 2004.
3. R. Koodli "Fast Handovers for Mobile IPv6," RFC 4068, July 2005.
4. H. Soliman *et al.*, "Hierarchical Mobile IPv6 mobility management(HMIPv6)," RFC 4140, August 2005.
5. K. Omae *et al.*, "Performance Evaluation of Hierarchical Mobile IPv6 Using Buffering and Fast Handover," Technical Reports of IEICE, IN2002-152, December 2002.
6. K. Omae *et al.*, "Hierarchical Mobile IPv6 Extension for IP-based Mobile Communication System," Technical Reports of IEICE, IN2001-178, February 2002.

7. J. Jeong *et al.*, "Improved Location Management Scheme Based on Autoconfigured Logical Topology in HMIPv6," ICCSA 2005, vol. 3480, pp. 291-300, May 2005.
8. H. Jung *et al.*, "A Scheme for Supporting Fast Handover in Hierarchical Mobile IPv6 Networks," ETRI Journal, vol. 27, Number 6, December 2005.
9. Y. Gwon *et al.*, "Scalability and Robustness Analysis of MIPv6, FMIPv6, HMIPv6, and Hybrid MIPv6 Mobility Protocols Using a Large-scale Simulation," 2004 IEEE International Conference on Communications, vol. 7, pp 4087-4091, June 2004.

On the Security of Attribute Certificate Structuring for Highly Distributed Computing Environments

Soomi Yang

The University of Suwon
Kyungki-do Hwasung-si Bongdam-eup Wau-ri san 2-2,
445-743, Korea
smyang@suwon.ac.kr

Abstract. For an efficient role based access control using attribute certificate in highly distributed computing environments, we use a technique of structuring role specification certificates. The roles are grouped and made them into the relation tree. It can reduce management cost and overhead incurred when changing the specification of the role. Further we use caching of frequently used role specification certificate for better performance in case applying the role. And for the global space reduction, we also consider the issue of tree normalization. In order to be scalable distribution of the role specification certificate, we use multicasting packets. In the experimental section, it is shown that our proposed method is secure and efficient.

1 Introduction

A role based access control using attribute certificates can provide more flexible and secure environments than that using only public key certificates[1,2]. And the use of attribute certificates can provide more adaptable scheme by the use of role specification certificates. Highly distributed environments usually need support of the authorization of resources at varying levels of access. Furthermore, it needs the interactions of highly collaborating entities to be secure. However, the distributed environments could not have any central or global control. Therefore for security of highly distributed environments, we distribute the role specifications according to the levels of access. It accords with the characteristics of the distributed environments and sometimes is inevitable. We distribute the privileges. In addition, we group roles of privileges, which is different from the typical methods which group subjects only[1,2].

2 Attribute Certificate Structuring Model

Specific privileges are assigned to a role name through role specification certificate. The level of indirection enables the privileges assigned to a role to be updated, without impacting the certificates that assign roles to individuals. We make a chain of role specification certificates. It forms a tree structure similar to the structure used for group key management[3]. We call the node that corresponds to the role specification certificate having child role specification certificates as role group. If the nodes are

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 530 – 533, 2006.
© Springer-Verlag Berlin Heidelberg 2006

distributed geographically, the performance enhancements gained when the chained role specifications should be changed are overwhelming. Furthermore the application overhead can be overcome by the use of caching. The descriptions of attribute certificate structuring model in subsections use the following notations.

R : the number of roles

G: the number of the lowest level role groups $G_{\max} = \sum_{i=1}^{R} C_i$

g_i : role group i

s_i : role specification certificate related to role group g_i

h : height of the tree structure starting from 0

2.1 Role Distribution

The distribution of updated role specification certificates makes use of the multicast communication. Let $F(l)$ ($0 \le l \le h$)be the frequency of the transmission of a role specification certificate s_i in order to be successfully delivered to all $W(l)$ receivers. The probability that one of these $W(l)$ receivers (say w) will not receive the updated role specification if it is transmitted once is equal to the probability of packet loss, p, for that receiver. Let F_w be the frequency of role specification transmissions necessary for receiver w to successfully receive the role specification certificate. Since all the packet loss events for receiver w, including replicated packet and retransmissions, are mutually independent, F_w is geometrically distributed. Thus, the average expected frequency of the role specification packet transmission can be computed as following:

$$E[F(l)] = \sum_{f=1}^{\infty} P[F(l) \ge f] = \sum_{f=1}^{\infty} (1 - (1 - p^{f-1})^{W(l)}) \tag{1}$$

We can compute $E[F(l)]$ numerically using Equation (5) by truncating the summation when the f^{th} value falls below the threshold.

2.2 Role Caching

For an application of the role specification certificate, $2*(h-l)$ packets should be successfully transmitted. It forms a path through the role specification tree from a requesting node to a node having requested role specification certificate. To improve the application of the role specification, caching scheme can be adopted. Let $G(l)$ be the frequency of the transmission of packets for the successful delivery of a role specification certificate s_i to a requesting node. If the probability of having cached role specification certificate is q, the average expected frequency of the role specification packet transmission can be computed as following from the similar induction of role distribution.

$$E[G(l)] = (1-q)\sum_{f=1}^{\infty} P[G(l) \ge f] = (1-q)\sum_{f=1}^{\infty} (1 - (1 - p^{f-1})^{2(h-l)}) \tag{2}$$

A cached role specification does not need any packet transmission. It holds some space of a requesting device. q can be easily computed through the Markov chain state transition inspection.

2.3 Tree Normalization

For global space reduction, we can normalize the role specification. In normalized role specification tree, privilege should appear only in one certificate. The number of certificates in level l is ${}_{G_{norm}}C_{h+1-l} = 2^l$. Thus the total number of certificates can be induced to $\sum_{i=1}^{G_{norm}} {}_{G_{norm}}C_i$. It can include unused role group. However it generally significantly reduces the space compared to the space used for naturally evolving and diminishing role specifications without any regulation. Furthermore it reduces communication cost incurred when changing the specification of the role as shown in Table 1.

Table 1. The number of packet transmissions for changing the specification of the role

	Un-normalized		normalized	
	ungrouped	grouped	ungrouped	grouped
$f = 10$, $p = 0.1$	2.54	1.39	1.66	1.21
$f = 200$, $p = 0.9$	45.53	20.27	26.30	14.74

However normalization increases path length and incurs more packet transmission when the subject is going to apply the role specifications. Therefore the normalization should be adopted with caching conforming to the characteristics of individual environment.

3 Performance Evaluation

We measure the expected number of packet transmission, $E[F(l)]$ and $E[G(l)]$, for the performance comparison. For each given caching ratio q, we can inspect the effects to the average packet transmission. The 'cached' case of Fig. 1 shows the frequency

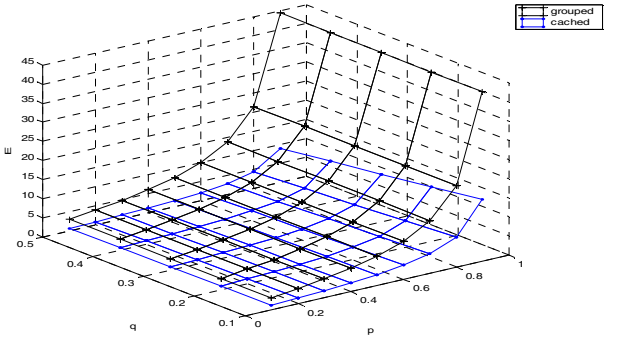


Fig. 1. A comparison of the expected packet transmission as a function of p and q

variations by packet loss p and caching ratio q with $f=100$. When the packet loss is small, the difference is small. However, as the packet loss gets bigger, it suffers more increasing packet transmission. Fig. 1 shows the plot of the expected packet transmission $E[F(l)]$ and $E[G(l)]$ for packet loss p and the caching ratio q. Fig. 1 shows the greater increase in $E[F(l)]$ where the roles are grouped than in $E[G(l)]$ where the roles are grouped and the role specifications are cached. If we take a specific sample case, from the values given in Fig. 1 we can see that total number of the packet transmission should be greatly decreased when the role specifications are cached. However if the nodes are distributed geographically and the packet loss is more often, the performance enhancements gained when the role specifications should be changed are overwhelming.

4 Conclusion

As an efficient access control using attribute certificate, we use the technique of structuring role specification certificates and reinforce it through caching them. It can reduce the management cost and overhead incurred when changing and applying the specification of the role. We grouped roles, made the role group relation tree, and showed the model description. It provides the secure and efficient role updating, applying and the distribution. For scalable role specification certificate distribution, we used multicasting packets. The performance enhancements are quantified with taking into account the packet loss.

References

1. ITI (Information Technology Industry Council), Role Based Access Control ITU/T. Recommendation X.509 | ISO/IEC 9594-8, Information Technology Open Systems Interconnection-The Directory: Public-Key and Attribute Certificate Frameworks (2003)
2. S. Farrell and R. Housley, An Internet Attribute Certificate Profile for Authorization, IETF RFC 3281, (2002)
3. Sandro Rafaeli, David Hutchison, A Survey of Key Management for Secure Group Communication, ACM Computing Surveys, Vol. 35, No. 3 (2003)

Performance Analysis of Single Rate Two Level Traffic Conditioner for VoIP Service

Dae Ho Kim, Ki Jong Koo, Tae Gyu Kang, and Do Young Kim

Electronics and Telecommunications Research Institute, DaeJeon, Korea
{dhkim7256, kjkoo, tgkang, dyk}@etri.re.kr

Abstract. In this paper, for the combination of DiffServ and RSVP, we propose traffic conditioning algorithm. Through the proposal, we expect the service quality improvement of real time application, especially voice traffic, by separation of other data traffic.

1 Introduction

In today's networking, the most interesting things are networking technologies based on a different kind of voice transmission called packet voice. But voice packet has the different characteristics with data packet. Voice packet is delay sensitive and does not tolerate dropped packets and retransmissions, compared with data packet. Minimization of delay and delay variance in network is required to improve the voice quality.

For this reason, in the Internet of today there is an ongoing discussion about realizing Quality of Services (QoS). One approach to achieve this was the development of the Resource Reservation Protocol (RSVP) [1]. The alternative concept for a QoS supporting Internet is the so called Differentiated services (DiffServ) [2]. Combination of the advantages of DiffServ (good scalability in the backbone) and of RSVP (application support) is needed.

At this combination process, we propose Single Rate Two Level Traffic Conditioner and related algorithm. Through the proposal, we expect the service quality improvement of real time application, especially voice traffic, by separation of other data traffic. This expectation is based on that voice traffic.

2 Flow-Based Rate Control Algorithm

2.1 Single Rate Two Level Traffic Conditioner

In this section, we propose Single Rate Two Level Traffic Conditioner which meters and shapes input traffic according to its reservation style in EF class of DiffServ ingress node. Single Rate Two Level Traffic Conditioner has two leaky buckets which one is for RSVP Fixed-Filter style traffic and the other for other RSVP reservation style traffic and just admission controlled traffic.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 534 – 537, 2006.
© Springer-Verlag Berlin Heidelberg 2006

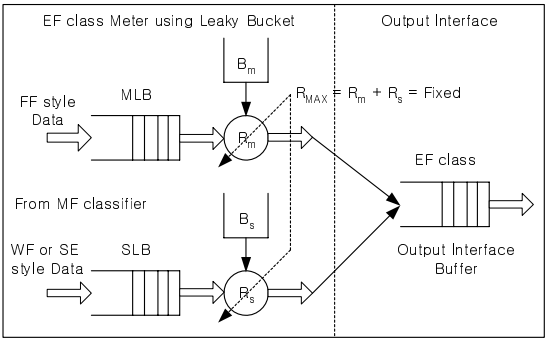


Fig. 1. Single Rate Two Level Traffic Conditioner architecture in the Ingress node of DiffServ network

Single Rate Two level Traffic Conditioner consists of two leaky buckets. We call these two leaky bucket, master leaky bucket and slave leaky bucket, because slave leaky bucket rate is controlled by master leaky bucket rate. Master Leaky Bucket (MLB) is reserved by RSVP Fixed-Filter (FF) style traffic. Its initial leaky bucket rate Rmi is 0 and is increased by RSVP FF style reservation (1). Slave Leaky Bucket (SLB) is same with general leaky bucket in DiffServ ingress node and used by RSVP WF and SE style reservation and just admission controlled traffic. Its initial leaky bucket rate R_{si} is EF class maximum rate R_{MAX} (1). R_{MAX} is the maximum amount of in-profile traffic in order for a network administrator to guarantee a high forwarding probability to that traffic.

$$R_{mi} = 0 = R_m \ , \ R_{si} = R_{MAX} = R_{s.} \tag{1}$$

$$R_m + R_s = R_{MAX} = \text{Fixed.} \tag{2}$$

Leaky bucket rate of MLB R_m is increased by RSVP FF style reservation request rate R_r , and leaky bucket rate of SLB R_s is decreased as a amount of increased rate of MLB R_r (3). Because the amount of in-profile traffic is fixed in DiffServ network, total leaky bucket rate must constant. So the sum of R_m and R_s is constant to R_{MAX} (4).

$$R_m' = R_m + R_{r.,} \quad R_s' = R_s - R_r. \tag{3}$$

$$R_m' + R_s' = R_{MAX} = \text{Fixed.} \tag{4}$$

Fig. 4 shows Single Rate Two Level Traffic Conditioner (TLTC) architecture in the Ingress node of DiffServ network. This TLTC must be used with MF classifier which classifies packets based on IP address and DSCP value.

2.2 System Architecture of RSVP Enabled DiffServ System

All these mechanisms are performed dynamically by RSVP soft state characteristics, that is, if destination indicates service quality degradation to source, source will

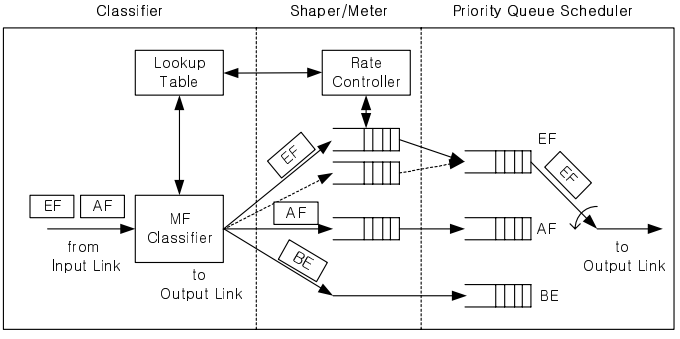


Fig. 2. System Architecture of DiffServ Ingress Router

request service quality guarantee to network elements with RSVP refresh message. Ingress node receiving RSVP Path message for refresh will update service parameters, bucket rate and size. Fig. 2 shows system architecture of DiffServ ingress router when our proposal is used.

3 Simulation Result

3.1 DiffServ Network

Fig. 3 shows network delay and delay variance of voice traffic of each class in DiffServ network. According to classification and priority scheduling of packet, high class packets get higher quality of service. But delay of EF class is higher when offered load of EF class increases more than 90% of limited in-profile traffic. This is effect of leaky bucket shaper. Fig. 3 shows also poor quality of EF class in delay variance aspect. In this result, we can know also importance of admission control of EF class and dynamic control.

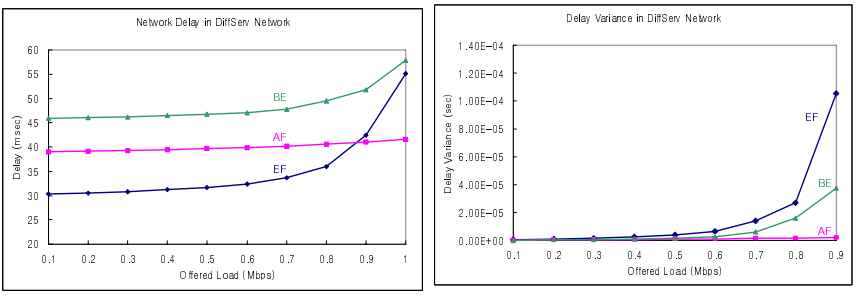


Fig. 3. Network delay and Delay Variance in DiffServ network

3.2 RSVP-Enabled DiffServ Network

Fig. 4 shows network delay and delay variance of voice traffic of each class in RSVP-enabled DiffServ network. Network delay of RSVP uncontrolled classes, EF, AF and BE, show same pattern with network delay in DiffServ network. But RSVP controlled class (RSVP EF in Fig. 4) shows high performance of low delay and delay variance.

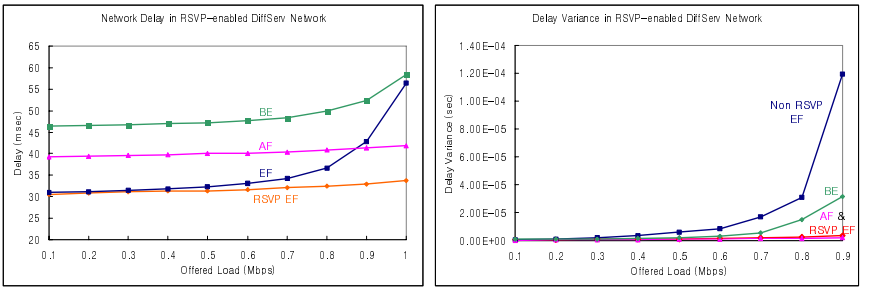


Fig. 4. Network Delay and Delay Variance of RSVP-enabled DiffServ network

4 Conclusion

Through this thesis and simulation, we can know importance of traffic admission control and dynamic rate control of shaper in DiffServ network. Decreasing effort of delay and delay variance in shaper of DiffServ ingress node must be studied and analyzed continuously. It could be obtained by RSVP mapping to DiffServ network, we think. So more detail study and research are needed.

References

1. R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, “Resource Reservation Protocol (RSVP) – version 1 Functional Specification,” RFC 2205, Sep 1997.
2. S. Blake, D. Black, M. Carson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services,’ RFC 2475, Dec 1998.
3. R. Balmer, F. Baumgarter, T. Braun, M. Gunter, “A concept for RSVP over DiffServ,” Computer Communications and Networks, Ninth International Conference on, p.p. 412 - 417, 2000.

An Architectural Framework for Network Convergence Through Application Level Presence Signaling

Atanu Mukherjee

Chief Architect
Cognizant Technology Solutions Corporation
Tel.: 973.368.9300
atanu.mukherjee@cognizant.com

Abstract. Over the past few years we have witnessed steady progress toward convergence in communications. We believe we are now witnessing convergence beyond just media transport. Further, abstract application level signaling technologies are poised for takeoff and will serve as the unifying fabric for multi-dimensional convergence. Our research shows that such a signaling mechanism known as "presence", combined with advances in namespace and security technology, will enable ultimate service innovations.

Keywords: presence management, SIP, application layer signaling, network convergence.

1 Convergence—The Challenge and Opportunity

The future of convergence in communications is the unification of, and interoperation across media, networks, devices, services and namespaces. Today, we are able to detect some of the key elements forming around the various evolving technologies and business models.

We have started to see evidence of progress in this phenomenon. Some network providers have begun to differentiate with value added services like universal registries, hosting, transaction based [1], and leveraging other services like overlay signaling [2] and namespace enablement [3], with a vision of even tighter integration of such services [4]. Abstract application level signaling through presence management has the power to unify networks, devices, services and namespaces to unleash service innovation that will have long-lasting impact.

2 Presence Signaling Drives Convergence

The best way to understand presence is to recognize that a person has a context, or state, at every given instant in time. Technically, this context can be captured and reflected through the devices, networks and applications with which a person interacts. This context can then define the information space and hence a person's service needs. A user's context changes over time and space and can be thought of as fractal iterations over a base context, or set of states. The figure below illustrates this concept of presence.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 538 – 541, 2006.
© Springer-Verlag Berlin Heidelberg 2006

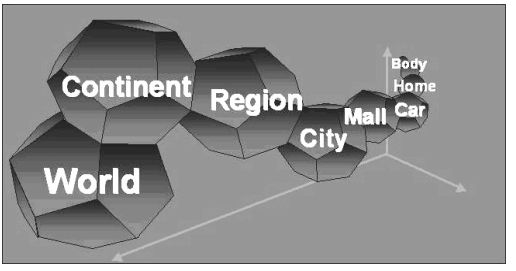


Fig. 1. User demands change in Time and Space

Presence - A Definition: More formally, "Presence" is defined as the subscription to and notification of changes in communication states of a user, where the communication states consist of a set of communication means, such as communication devices, location, willingness to communicate, preferences, and the personal state of the user.

Presence management technology captures the dynamically changing state of the user as evinced by the end user's interaction with the devices, applications and the network, abstracts that state information using a simple and easy to understand mechanism, and disseminates it to the subscribing application/services, based on an opt-in model. The richness of presence information consumed by an application depends on the number and precision of attributes that the network is capable of capturing and reporting between the two ends of a potential communications session.

The key principles that drive a robust application level presence signaling architecture are:

- **Automatic And Dynamic Capture Of Presence –** To the extent state information can be automatically captured by a presence management solution, presence becomes meaningful, accurate and rich. This requires that the presence management software be able to capture state information from devices and the network. Automatic and dynamic capture of presence information across multiple networks, device types and proxies is the key differentiator for the rich convergence applications and services of the future.
- **Event Based Signaling Model –** Presence must be captured and disseminated using an event-driven asynchronous design, rather than a query based model. An event-driven publication and subscription model provides accurate updates in near real-time and has the capacity to provide "carrier class" accuracy. The presence information can then be queried by an application, like a naming service discussed earlier, with updates in the presence servers triggered by events.
- **Abstract Signaling –** Presence information must be abstracted to be of value to a broad set of applications and services. To create the proper abstraction, activities in devices and networks must be correlated with other attributes, like location,

preferences and willingness to communicate, and this aggregated compilation of state must be signaled to the end application or service.

- **Simplicity –** Applications and services must be able to consume presence information in a simple way. Presence payload definitions using XML signaled over SIP is an elegant and simple way for transporting presence information.
- **Scaling –** Capture of presence transactions, and the subsequent processing and dissemination, must scale to multiples of millions in near real-time. This presents challenges in the design of the software, which must be architected and designed using the principles of distributed systems for carrier class scaling.
- **Fine Grained Access Control –** Dissemination of presence information, by definition, must follow an opt-in model. But that alone is not enough. An end user must be able to specify who has access to what level of the user's state information in a simple intuitive way. [5]

A Framework for Presence Signaling

The following schematic outlines the high level architecture of presence signaling

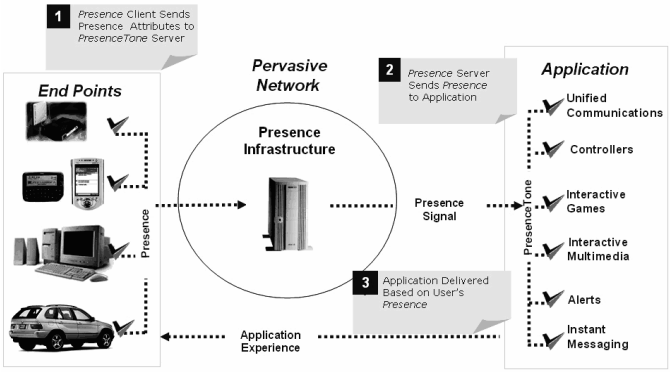


Fig. 2. Using SIP to Signal Presence

3 Conclusion

In conclusion, the business of offering abstract application signaled presence-based converged services is the next big opportunity in global communications. Enabling such an environment with seamless real-time communication and massively secure scalable interaction, while fostering open innovation, requires underlying middleware architecture. The presence architectural framework lays the groundwork for enabling such pervasive converged networks and provides the platform to create new product and service innovations for next generation carriers and service providers. Limitations exist only in the imagination and only limited vision can thwart the growth and ultimate success of the emerging converged communications industry.

References

1. http://www.verisign.com
2. http://www.illuminet.com
3. http://www.netnumber.com
4. “Verisign Plans to Pay $1.2 B in stock for Illuminet Holdings”, Wall Street Journal, 26th September, 2001
5. Private Escrow Key Management : A Method and Its Issues, Edward M Scheidt, TECSEC, Key Escrow Issues Meeting, NIST, Gaithersburg, Maryland, Sept., 6, 1995

Security Approaches for Cluster Interconnection in a Wireless Sensor Network

Alexandre Gava Menezes[1] and Carlos Becker Westphall[2]

[1] Data Processing Center – Federal University of Santa Catarina
NPD – Campus Universitário – 88040-970, Florianópolis, SC, Brazil
[2] Network and Management Laboratory – Post-Graduate Program in Computer Science
Federal University of Santa Catarina
Caixa Postal 476 – 88040-970, Florianópolis, SC, Brazil
shakal@npd.ufsc.br, westphal@lrg.ufsc.br

Abstract. A wireless sensor network is a collection of devices limited in low-powered batteries, processing, communication bandwidth capabilities and low memory availability. Due to these constraints, most of security approaches used in wired networks cannot be applied directly in this environment. In this work, we present a hybrid protocol that treats the group key management scheme and the transparent cluster interconnection. The feasibility of this protocol was verified by simulation and we concluded that increasing the number of nodes in the network does not change the performance of the interconnection between any two clusters.

1 Introduction

A wireless sensor network is characterized by a collection of nodes, which do not utilize a network infrastructure. The low processing capability, low memory availability, and the battery-based energy consumption are also limiting factors that should be taken into consideration. In this sensor network environment, due to the limitation on the transmission range, in order for two nodes, A and B, to communicate, they must utilize multi-hop routing over intermediate nodes between them. In this sense, nodes act not only as hosts, but also as routers, receiving and transmitting messages whose end-destinations are other nodes.

Because of the nodes limitations, the utilization of a pure public-key infrastructure (PKI) becomes unviable because of the processing time and consequently because of the energy consumption necessary for the encryption and decryption of messages. One way to reduce the processing time and the energy consumption in message passing is the utilization of symmetric keys, both in routing and data exchange. A sensor network can be divided in clusters, in the attempt to localize group sensors by the characteristics of the services they use and provide, with the goal of optimizing performance [1].

In this article we present a hybrid secured protocol which guarantees group key establishment, route discovery and secure message passing. The feasibility of this protocol will be proved by a simulation. We simulate an environment of a variable number of clusters conclude that a significant increase in the number of nodes in the

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 542 – 545, 2006.
© Springer-Verlag Berlin Heidelberg 2006

network does not change the performance of the interconnection between any two clusters. Our discussion focus in section II we will present the proposed protocol we will also present the transparent interconnection of clusters. We present in section III the previous simulations and the results obtained, and in section IV we conclude the article.

2 The Proposed Protocol

In this article, we propose a hybrid protocol that combines the advantages of the use of symmetric keys in the discovery of routes and in the message exchange between sensors and public keys for the establishment of symmetric keys, as well as transparent cluster interconnection.

In order to turn the use of public keys in a sensors network viable, it is preferred that only one node per cluster utilizes a CA to authenticate the received public keys. The node must be the cluster head, due to the less constrained characteristics, so that the solution will need fewer distributed CAs to ensure the availability to the entire network. To keep the focus of this article, we assume that there is a CA distribution scheme that allows a cluster head to verify the authenticity of another node in which it wants to communicate. To turn the use of a cryptography protocol viable, the sensors will establish a session key (symmetric) with the cluster head and once this key is obtained, all route discovery operations and message exchange will be done utilizing this group key. In this approach, the cluster head becomes the entity responsible by the key management and distribution.

The key establishment scheme follows some basic rules. Each sensor, before entering the network, has a set of pre-installed data by a trusted entity. These values are its own sensor ID (IDsns) and a pair of keys , one private (KRsns) and another public (KUsns). The sensor also stores the identification (IDch) and the public key (KUch) of the cluster head in which it is supposed to be placed. This approach is justified because of the cost to generate the pair of keys and to establish a trust relation between the sensor and the cluster head will be higher.

The messages used to the key establishment are:

$$\text{Sensor} \rightarrow \text{CH: (type, IDch, IDsns, KUsns, Signature)} \quad (1)$$

$$\text{CH} \rightarrow \text{Sensor: (type, IDsns, IDch, EKUsns(Ks), Signature)} \quad (2)$$

$$\text{CH1} \rightarrow \text{CH2: (type, IDch2, IDch1, IDsns, KUsns, IDsns1, Signature sns, Signature ch1)} \quad (3)$$

$$\text{CH2} \rightarrow \text{CH1: (type, IDch1, IDch2, IDsns, Signature ch2); with Signature ch2 = Sign KRch2(HASH(type, IDsns, IDch2, IDch1, KUch1, IDsns1))} \quad (4)$$

$$\text{CH1} \rightarrow \text{Sensor: (type, IDsns, IDch2, IDch1, KUch1, IDsns1, Signature ch2)} \quad (5)$$

$$\text{Sensor} \rightarrow \text{CH: (tipo, IDch, IDsns, Signature gr); with Signature gr = EKs(Hash)} \quad (6)$$

Where (1) is the request to join the cluster and obtain the group key. (2) is the response to the join request and transmission of the group key. (3) is the requests forward to join the cluster to the destination cluster head. (4) is the configuration response of the sensor to accept the new cluster head. (5) is the reconfiguration message of the sensor to belong to the new cluster head. (6) is the request to obtain a new group key.

In this article, we are focused in avoiding the processing overload and energy consumption of the sensors. The idea is to transfer the key agreements to the cluster heads. This way, the processing time to establish a session key will be lesser if it is established for any two entire clusters instead of being established for just two sensors. If a sensor wants to communicate with another sensor inside the cluster, it simply does it by using the group key. When the destination sensor is in another cluster, the entire message path will be done using different session keys. First of all, the originator sensor encrypts the data messages using its own cluster group key. The cluster head uses the session key that was established with the destination cluster head. The destination cluster head uses its own group key. In the last step, the destination sensor accesses the secured information with the group key.

One of the main advantages of this approach is achieved when any other sensors of any of these two clusters want to communicate. The entire route discovery will be optimized due to the fact that the most complex process has already been done.

This way, we have a transparent cluster interconnection, because the sensor doesn't need to know in what cluster is the sensor that it wants to communicate to. This approach will be the base of our simulations to demonstrate that by increasing the number of the sensors and consequently the number of clusters in the network, the messages exchange performance will be the same, having only the increasing of the propagation time of the message through the neighboring nodes.

3 Simulations

The simulations had their main focus on the proposed protocol performance for the cluster interconnection, being necessary the creation of an environment in the Network Simulator-2 (NS-2). As the presented solution must be implemented in the routing layer, the Ad-hoc On demand Distance Vector (AODV)[2] protocol was used, with the SAODV (Secure AODV)[3] security requirements.

In our simulations, we simulate the 512 bytes CBR (Constant Bit Rate) messages being sent through any two sensors of any two different clusters, having each one 250 sensors and these sensors were distributed evenly inside the cluster. In this work we did not consider the mobility of the sensors. We considered frequencies beyond 8MHz, like the 16MHz, 32MHz and 64MHz frequencies.

The previous simulations proved the expected performance of the proposed protocol. In the Fig. 1, all four curves have a small angular coefficient, despite the fact that they seem have a bigger inclination curves from the tenth cluster. In the collected values, we can notice a gradual increase of the time due to the increasing number of clusters.

Increasing the number of clusters, the time for the message to reach the destination also increases, because the distance increases in the same proportion as the number of clusters. The propagation time is more relevant than the time to calculate the hash function.

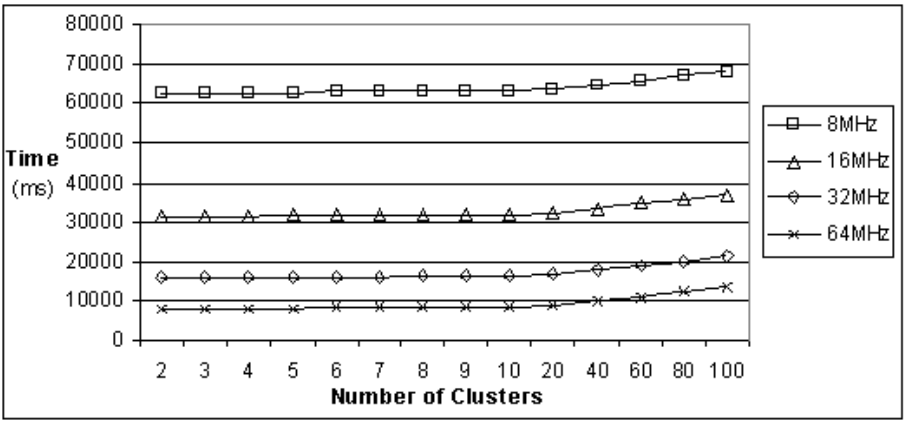


Fig. 1. The average time to send a CBR message of 512 bytes over the number of clusters in the network without any pre-established route

4 Conclusions

The present work tried to adopt the AODV protocol to a clustered sensors network environment. A key establishment and a mechanism to interconnect securely any two clusters was proposed. Analyzing the simulations results, we can notice an even performance in the clusters interconnection, even when the number of clusters was considerably increased. Some disadvantages of this approach must be exposed. The way the messages pass through can overload the cluster head, becoming a bottleneck to the extra cluster communication. Another point that can be a disadvantage is the way that the message authentication between intermediate cluster heads are treated, where there can be a delay to identify an anomalous message. However, we do believe that the benefits pointed out in this work are bigger than those casual disadvantages. The main attractive point of this approach is to have a constant performance in a message exchange between any two clusters, even with the increasing of the number of clusters in the network.

References

1. Bechler, M., Hof, H. J., Kraft, D., Pählke, F. e Wolf, L. (2004) “A cluster-based security architecture for ad hoc networks”. IEEE INFOCOM.
2. Perkins, C. e Belding-Royer, E. (2003) “Ad hoc on-demand distance vector (AODV) routing”. IETF Request for Comments, RFC 3561. July.
3. Zapata, M. G., e Asokan, N. (2002) “Securing ad hoc routing protocols”. Proc. ACM Workshop on Wireless Security (WiSe), ACM Press, pp. 1-10.
4. Kamal, A. B. M. (2004) “Adaptive Secure Routing in Ad Hoc Mobile Network”. Master of Science Thesis. Royal Institute of Technology. Sweden. November.

A Resource-Optimal Key Pre-distribution Scheme with Enhanced Security for Wireless Sensor Networks

Tran Thanh Dai, Al-Sakib Khan Pathan, and Choong Seon Hong*

Networking Lab, Department of Computer Engineering, Kyung Hee University, Korea
{daitt, spathan}@networking.khu.ac.kr ,cshong@khu.ac.kr

Abstract. This paper proposes an efficient resource-optimal key pre-distribution scheme for providing improved security in wireless sensor networks. We exploit the advantages of two schemes with some modifications and combine our encoding scheme to get resource-efficiency and better security.

1 Introduction

In this paper, we present a new key pre-distribution scheme utilizing the advantages of two existing schemes [1] and [2]. Our analysis shows that our combined scheme performs better than each of these two schemes and ensures enhanced security than most of the other existing schemes. Principal contributions of our work are: (1) Considerable improvement in sensors' resource usage while keeping security as the top priority (2) Rigorous guarantee of successfully deriving pairwise keys that enable node-to-node authentication and (3) Better network resilience as compromising one or more pairwise keys does not influence the remaining pairwise keys

2 Our Proposed Scheme

The building blocks of our scheme are [1] (referred to as *LU decomposition scheme* here) and [2]. We modified both of the schemes to adapt them with our scheme.

LU Decomposition Scheme [1]. Firstly, a large pool of keys P with size *s* is generated along with their identifiers. Then an $N \times N$ symmetric key matrix M is generated where *N* is the maximum number of sensor nodes that could be deployed. Each element M_{ij} of M is assigned a distinct key from the key pool P such that $M_{ij} = M_{ji}, i, j = \overline{1, N}$. LU decomposition is applied to matrix M in the next step. The results of this decomposition are two matrices L, the lower triangular matrix and U, the upper triangular matrix; both of which are $N \times N$ matrices such that M = LU. Now, every sensor node is randomly assigned one row from the L matrix and one column from the U matrix, following the condition that, the *i*th row of L, $L_r(i)$ and the

* This work was supported by the MIC and ITRC projects. Dr. C. S. Hong is the corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 546 – 549, 2006.
© Springer-Verlag Berlin Heidelberg 2006

*i*th column of U, $U_c(i)$ always go together when assigned to a sensor node. Now, let us explain how a common key could be found between two nodes. Assume that sensor S_i and sensor S_j contains $[L_r(i), U_c(i)]$ and $[L_r(j), U_c(j)]$ respectively. When S_i and S_j need to find a common secret key between them for communication, they first exchange their columns, and then compute vector products as: S_i: $L_r(i) \times U_c(j) = M_{ij}$ and S_j: $L_r(j) \times U_c(i) = M_{ji}$. As M is the symmetric matrix, definitely, $M_{ij} = M_{ji}$. M_{ij} (or M_{ji}) is then used as a common key between S_i and S_j.

Modified Blom's Symmetric Key Generation Scheme [3]. In this scheme, as long as no more than λ nodes are compromised, the network is perfectly secure (this is referred to as the λ-secure property). Increasing λ results in greater network resilience but also results in higher memory usage within each sensor node. During the pre-deployment phase, a $(\lambda+1)\times N$ matrix (where N is the maximum number of sensor nodes in the network) G over a finite field GF(q) and λ (the security parameter discussed earlier) are constructed. G is considered as public information; any sensor can know the contents of G, and even adversaries are allowed to know G. In order to achieve the λ-secure property any $\lambda + 1$ columns of G must be linearly independent. Let p be a primitive element of GF(q) and N < q. Then, each nonzero element in GF(q) can be represented by some power of p, namely p^i for some $0 < i \le q-1$. A feasible G can be designed as follows:

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ p & p^2 & p^3 & \cdots & p^N \\ p^2 & (p^2)^2 & (p^3)^2 & \cdots & (p^N)^2 \\ & & \vdots & & \\ p^\lambda & (p^2)^\lambda & (p^3)^\lambda & \cdots & (p^N)^\lambda \end{bmatrix}_{(\lambda+1)\times N}$$

Before deployment, a random $(\lambda+1)\times(\lambda+1)$ symmetric matrix D over GF(q) is computed and used to compute an $N\times(\lambda+1)$ matrix A, which is equal to $A=(D.G)^T$, where $(D.G)^T$ is the transpose of $(D.G)$. Matrix D needs to be kept secret and should not be disclosed to adversaries or any sensor node. As D is symmetric, it is easy to see: $K = A.G = (A.G)^T$. Thus, $K_{ij} = K_{ji}$, where K_{ij} is the element in K located in the *i*th row and *j*th column. In practice, G can be created by the primitive p of GF(q). Therefore, when storing the *k*th column of G at node k, it is only necessary to store the seed p^k at this node, any node can generate the column given the seed. After deployment, two sensor nodes *i* and *j* can find the pairwise key between them by exchanging their columns of G and using their private rows of matrix A to compute, $K_{ij} = K_{ji} = A(i).G(j) = A(j).G(i)$ where A(i) and G(j) represent the *i*th row of A and *j*th column of G respectively.

As we stated earlier that, our scheme is a combination of the two above-mentioned schemes with significant modifications so that it could be more apposite for the memory-constrained trait of wireless sensor networks. In our scheme, we only store the keying information in the sensor nodes, which are eventually used to derive two key halves to constitute a secret pairwise key. When two nodes want to communicate and are within the communication ranges of each other, the first key half is generated by

the LU key decomposition scheme while the second half is generated by modified Blom's key computation scheme. The pairwise key used to secure the communication link between two nodes is derived based on these two halves, i.e., a concatenation of the two halves or using one way hash functions (Figure 1).



Fig. 1. Components of a pairwise key

Our scheme consists of mainly two phases:

Prior Deployment: Key Distribution Phase. Let us consider that each sensor node has a unique id S_i where i=1,2,3…N. In our scheme, LU key decomposition scheme is applied first. After this step, each node S_i contains one row from the L matrix and one column from the U matrix, respectively. While storing the row and column information, we apply an encoding scheme. As in each row and column there are two portions (first part non-zero and then possible zero element part), to store one row of L and one column of U in each sensor, we only store the nonzero portions of them and one value specifying the number of following zeros in the zero-element part. Thus, we store the keying information for the first half of the target key instead of storing the full length of the key. This storing method is very effective when the size of the network is very large. For generating the other-half of the key, other information are assigned using Blom's key generation scheme. So, the matrices G and A are generated offline. In this case, column G(j) is assigned to node j. Although G(j) consists of $(\lambda+1)$ elements, each sensor only needs to memorize only one seed, which could be used to regenerate all the elements in G(j). We also store the jth row of A at this node. This information is secret and should stay within the node. A(j) consists of $(\lambda+1)$ elements. Therefore, for this case, each node needs to store $(\lambda+2)$ elements in its memory. As the length of each element is similar to the length of the second key-half of the pairwise key, the memory usage of each node is $\frac{\lambda+2}{2}$ times the length of the key.

After Deployment: Pairwise Key Establishment Phase. After deploying the sensors on the area of interest (AOI), when two nodes (let S_i and S_j) want to communicate with each other, they need to establish a pairwise secret key to transmit data securely. The steps that are followed for this are:

(1) S_i and S_j use some mechanisms to discover the location of the other (e.g., using query broadcast messages)
(2) S_i and S_j exchange messages containing $U_c(i)$, $U_c(j)$ from matrix U and a seed to generate, G(i), (G(j)) from matrix G
(3) S_i and S_j then compute the first half of the pairwise key as follows: $S_i : L_r(i) \times U_c(j) = M_{ij}$ and $S_j : L_r(j) \times U_c(i) = M_{ji}$
(4) S_i and S_j compute the second half of the pairwise key using the following equation: $K_{ij} = K_{ji} = A(i).G(j) = A(j).G(i)$

Up to this step, both S_i and S_j have the two halves of the pairwise key. To derive the pairwise key, the simplest way is to concatenate the first half with the second half. The pairwise key can also be created by using a one-way hash function. This key is stored in the memory of the two sensors for the rest of their communication. In our scheme, to reduce energy overhead for encryption and decryption of information exchanged between a pair of non-neighboring sensor nodes, each sensor node has a message relaying function. That is, messages exchanged between two non-neighboring nodes are encrypted and decrypted only by the concerned nodes using a common pairwise key established as described in the steps. The intermediate nodes only have to relay the messages to the receiving node. They do not need to understand the contents of the messages. Hence, they do not need to encrypt or decrypt the messages, which saves the computation and energy power.

3 Results and Conclusions

Here, we have presented a new pairwise key pre-distribution scheme for wireless sensor networks. Due to the page constraint, we have omitted the analysis part and shortened the details of our scheme. Our scheme is scalable because sensor nodes do not need to be deployed at a time rather they could be added later, and still they will be able to derive secret keys with existing nodes. Also the network-wide memory that is saved in our scheme could be used for network-wide distributed tasks. In Figure 2 we only show the storage efficiency that is gained from the encoding in our scheme.

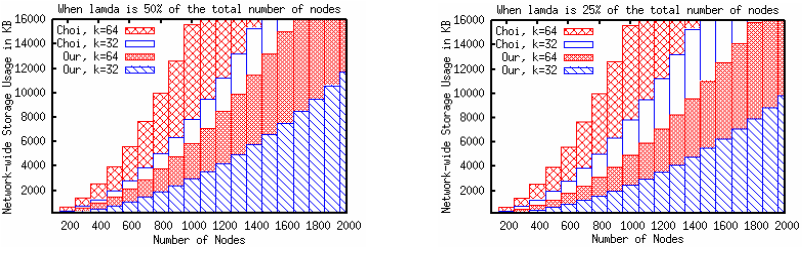


Fig. 2. Network-wide Memory usage in our scheme and [1] (left) if λ is 50% of the total no. of nodes (right) if λ is 25% of the total no. of nodes [λ could be smaller]

References

[1] Choi, S. and Youn, H., "An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks", EUC Workshops 2005, LNCS 3823 (2005) 1088-1097.

[2] Blom. R., "An optimal class of symmetric key generation systems", Advances in Cryptology: Proceedings of EUROCRYPT 84 (Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, eds.), Lecture Notes in Computer Science, Springer-Verlag, 209 (1985) 335-338.

[3] Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., and Khalili, A., "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks", ACM Transactions on Information and System Security, Vol. 8, No. 2, May (2005) 228-258.

Intelligent Home Network Service Management Platform Design Based on OSGi Framework

Choon-Gul Park, Jae-Hyoung Yoo, Seung-Hak Seok,
Ju-Hee Park, and Hoen-In Lim

[1] IP Network Research Department, Network Tech Lab, Korea Telecom,
463-1, Jeonmindong, Daejeon, Korea
{ktgen, styoo, suksh, jhpark78, limhi}@kt.co.kr

Abstract. In this paper, we propose an open standard-based, intelligent service management platform that aims to provide a flexible and extensible platform for building intelligent home network services. To have greater flexibility and efficiency in managing home network services, we further propose an AAA proxy, a service provisioning, security agent, a sip proxy service, and others. We call this platform the IHSM (Intelligent Home Network Service Management) Platform, which consists of the ISMS (Intelligent Service Management System) and the ISMF (Intelligent Service Management Framework).

1 Introduction

Along with the explosive growth of broadband access and Internet services, networked home appliances and devices are also increasing, and the expectations on various home network services are growing. Because of this situation, the SP (Service Provider) is now conceptualizing the Blue Ocean that makes the competition irrelevant by offering various home network services and fulfilling the customer's expectations of a better quality of life through intelligent home network services.

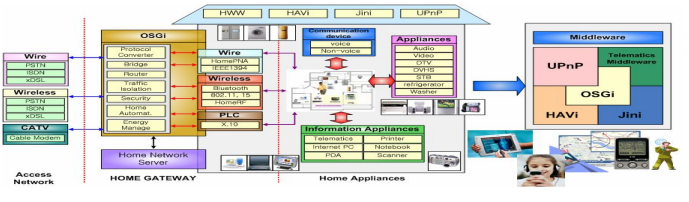


Fig. 1. The infrastructure and base technologies for the intelligent home network service [1]. The elements of home network technology described in the figure are the home gateway, home gateway middleware and home appliances access technologies.

As you can see in Fig. 1, it is necessary to use a fusion of technologies to provide intelligent home network services., To provide an intelligent home network service, a service provider needs an open architecture-based middleware that can satisfy the requests of various types of home network services faster and more flexibly.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 550 – 553, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In the past 3-5 years, various researches based on the OSGi Framework have been done[2][3][4][5]. Haitao Zhang presented a mobile-agent and OSGi-based three-tier control sys-tem architecture for a smart home [6]. Akihiro Tsutsui suggested a Management Ar-chitecture and Distribution Framework for Home Network Services at an NGN work-shop in 2005 [7]. Daqing ZHANG presented an OSGi-Based Service Infrastructure for Context Aware Automotive Telematics [8]. Xie Li proposed an OSGi-based home network system [9].

In this paper, we propose an open standard-based, intelligent service management platform that aims to provide a flexible and extensible platform for building intelligent home network services. To have greater flexibility and efficiency in managing home network services, we further propose an AAA proxy, service provisioning, security agent, sip proxy service, and others.

The rest of the paper is organized as follows. First, the overall technologies for meeting the management requirements of intelligent home network services are introduced in Section 2. Then, our IHSM platform and details are presented in Section 3. Finally, some concluding remarks and future works are drawn for the paper in Section 4.

2 Management Requirements for Intelligent Home Network Services

The concept of an open structure for the home network enables home network service providers to accept both existing services and new services at the same time through a common service platform. In addition, we can construct an infrastructure that is able to manage these services effectively through this concept.

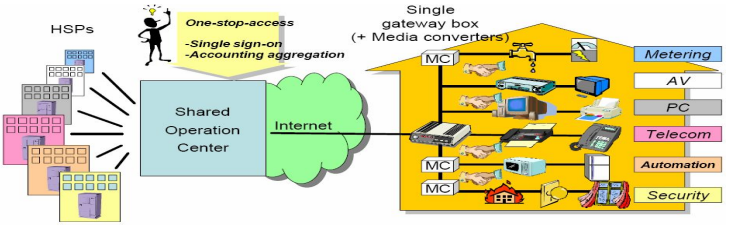


Fig. 2. Open service structure enables home network service providers to accept both existing services and new services at the same time through a common service platform [7]

Integrated management technology is one of the most important constituent technologies for offering an intelligent home network service. That is, technology development and architecture design that can integrate base technologies such as middleware, wired/wireless network, home gateway, home server, and others are important.

On the other hand, considering a home network environment with various networks and devices, an intelligent middleware technology is required to support interoperability among these devices, automated configuration management, multimodal interaction, and QoS. We also need knowledge-based service middleware technologies for intelligent

personalized services and service management technologies to provide more advanced services continuously.

In addition, a security technology that protects the private information of users and can quarantine the security weakness of the network is also important.

Accordingly, we needed a service management platform based on OSGi, which is an open standard for the management of an effective and flexible intelligent home network service. Thus, we designed a management platform focusing on the requirements on the constituent technologies, such as service management technologies, technologies for integrating the home network architecture, privacy and security management, etc.

3 Intelligent Home Network Service Management Platform

The proposed intelligent home network service management (IHSM) platform consists of the intelligent service management framework (ISMF) and the intelligent service management system (ISMS), which were designed for an OSGi-based home gateway. The ISMF was designed based on the service provider's requirements for managing the intelligent home network while the ISMS was designed based on the SMF.

The architecture of the IHSM platform is shown in Fig. 4. As shown in Fig. 4, the ISMF is the main framework of the IHSM and it includes the OSGi framework, which demonstrates the service proposed at OSGi specification. That is, the ISMF adds the AAA proxy, service provisioning, management agent, and security agent on the OSGi framework. Based on this ISMF, we aim to offer session initiation protocol (SIP) proxy and home network security services. More explanation about the modules of the ISMF follow in the next paragraphs[10][11].

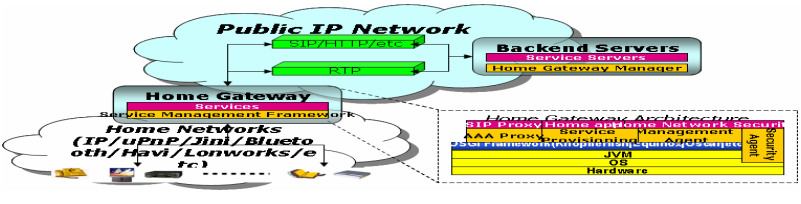


Fig. 3. The intelligent home network service management (IHSM) platform consists of the intelligent service management framework (ISMF) and the intelligent service management system (ISMS), which were designed for an OSGi-based home gateway

4 Conclusion and Future Work

In this paper, we propose and design an open standard-based, intelligent service management platform and architecture. By analyzing the OSGi framework and the runnable mechanism of bundles and integrating them with the requirements of intelligent home network services, we designed a management platform that aims to provide a flexible and extensible platform. Its features are:

- An AAA proxy is designed based on unified authentication for service management focusing on customers. The user receives authentication through the virtual AAA server and the AAA proxy.
- The management agent enables the user or operator to manage the home gateway and services remotely.
- The process of service provisioning. Typically, the customer doesn't have any knowledge about the provisioning of devices.
- The repository manager can carry out and manage operations such as registration, update, search and delete on the service bundles of the home devices at the back-end servers.
- The SIP proxy service bundle plays a role in routing messages between the back-end SIP server and the SIP UA.
- The IP network security framework quarantines many kinds of harmful traffic, manages security bundles that were designed to protect systems from worms and viruses, and manages other functions for network security.

We will continue to design and implement more details and make it practical. Our major research areas in the future are on context aware-based service management and home network security management.

References

1. National Computerization Agency: "The present and direction of digital home business", Digital home service model excavation workshop, (2003)
2. Kyu-Chang Kang, Jeon-Woo Lee: "Implementation of Management Agents for an OSGi-based Residential Gateway", Embedded S/W Technology Center, Computer and Software Lab, ETRI(2003)
3. Choonhwa Lee, David Nordstedt, Sumi Held: "Enabling smart spaces with OSGi, Pervasive Computing IEEE, Volume-2 Issue-3, July-Sept, ZW3, pp.89-94 (2003) pp.89-94.
4. Open Service Gateway Initiative, About the OSGi Service Platform, Technical Whitepaper Revision 4.1", http: www.osgi.org (2005)
5. Richard S. Hall and Humberto Cervantes: "An OSGi Implementation and Experience Report", Laboratoire LSR Imag, rue de la Chimie Domain Universitaire, France(2004)
6. Haitao Zhang, Fei-Yue Wang, Yunfeng Ai: "An OSGi and Agent Based Control System Architecture for Smart Home", IEEE(2005)
7. Akihiro Tsutsui: "Management Architecture and Distribution Framework for Home Network Services", NGN Workshop(2005)
8. Daqing ZHANG, Xiao Hang WANG: "OSGi Based Service Infrastructure for Context Aware Automotive Telematics"(2004)
9. Xie Li, Wenjun Zhang: "The Design and Implementation of Home Network System Using OSGi Compliant Middleware", IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, IEEE(2004)
10. Session Initiation Protocol(SIP), http://www.ietf.org/rfc/rfc2543, (1999)
11. Session Initiation Protocol(SIP), http://www.ietf.org/rfc/rfc3261, (2002)

COPS-Based Dynamic QoS Support for SIP Applications in DSL Networks

Seungchul Park[1] and Yanghee Choi[2]

[1] School of Internet Media Engineering, Korea University of Technology and Education
Byeongcheon-myun, Cheonan, Chungnam, Republic of Korea
scpark@kut.ac.kr
[2] Department of Computer Engineering, Seoul National University
Shinlim-dong, Kwanak-ku, Seoul, Republic of Korea
yhchoi@snu.ac.kr

Abstract. In this paper, several dynamic QoS solutions including Direct DiffServ, Admission-based Direct DiffServ, Indirect DiffServ, and Hybrid DiffServ are proposed to support largely emerging SIP-based P2P(Peer-to-Peer) and ASP(Application Service Provider) multimedia applications in DSL networks, most widely deployed as broadband access networks. The proposed solutions are designed to be based on standard COPS protocol which is simple and service-independent.

1 Introduction

SIP-based multimedia applications can be serviced in P2P(Peer to Peer) environment, where each end user is responsible for the QoS provisioned, as well as in ASP(Application Service Provider) environment, where the ASP is responsible for the QoS. Dynamic QoS for a SIP multimedia application needs to be supported in an appropriate way according to the corresponding service environment of the application. Currently most QoS-enabled access networks are developed based on the DiffServ IP QoS architecture because of the complexity problem of the other IntServ QoS architecture[1,2,3]. In this paper, several dynamic QoS solutions including Direct DiffServ, Admission-based Direct DiffServ, Indirect DiffServ, and Hybrid DiffServ, are proposed to support largely emerging SIP-based multimedia applications in DSL networks. How to apply each proposed dynamic QoS solution to the corresponding service and network environment is also discussed in this paper.

2 Proposed Solutions: Direct DiffServ, Admission-Based Direct DiffServ, Indirect DiffServ, and Hybrid DiffServ

1) Direct DiffServ solution for dynamic QoS support enables end-user entities of a SIP multimedia application to directly access the corresponding IP QoS required for the application's dynamic QoS. In the Direct DiffServ solution, UAC(User Agent Client) and UAS(User Agent Server) firstly collect the identification information and QoS attribute values for each media stream of a multimedia application through SDP

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 554 – 557, 2006.
© Springer-Verlag Berlin Heidelberg 2006

offer and SDP answer exchange procedure embedded within SIP session establishment[4,5], and perform DiffServ classification based on the collected QoS data. The DiffServ classification is an environment-specific matter. (a) of Fig.1 shows an example scenario of Direct DiffServ dynamic QoS solution for a simple SIP multimedia application which does not include QoS preconditions.

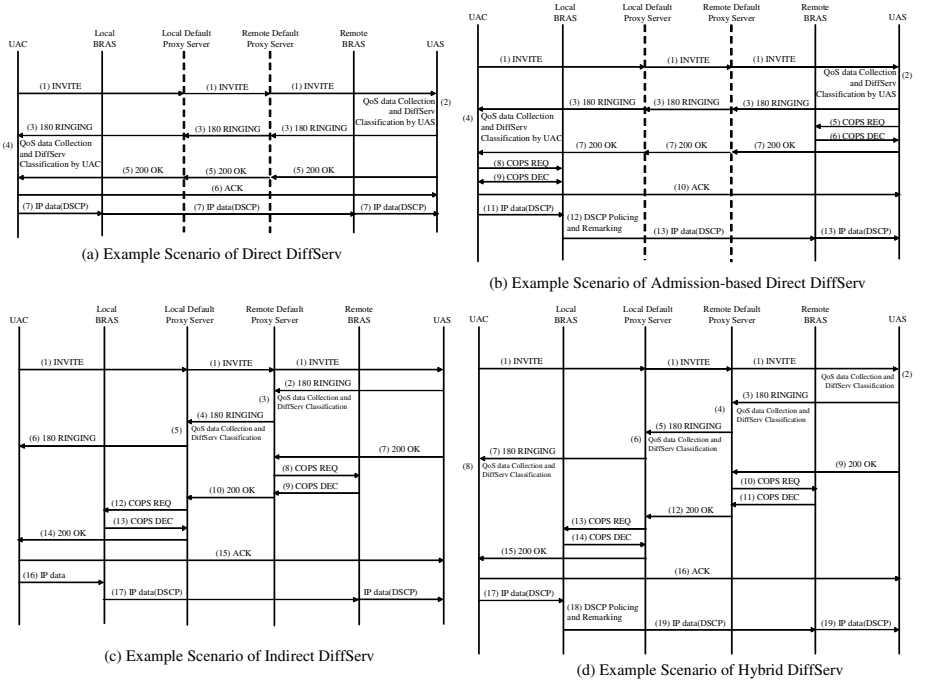


Fig. 1. Example scenarios of Direct DiffServ, Admission-based Direct DiffServ, Indirect DiffServ, and Hybrid DiffServ

Since Direct DiffServ does not require any additional signaling to support dynamic QoS, it is very simple. But it does not provide any mechanisms to do admission control for IP QoS and policing service of the admitted IP QoS to protect authorized users from non-authorized users. This means that it is difficult to apply Direct DiffServ solution to the most broadband access environments where QoS policing is necessarily required to filter non-authorized QoS packets. We believe that Direct DiffServ solution will be useful to support P2P SIP multimedia applications, in which end systems are fully responsible for QoS support, in enterprise network environments where end users who are sending QoS packets can be trusted.

2) Admission-based Direct DiffServ solution adds some signaling mechanism between end-users and NSP to the Direct DiffServ solution. Through the signaling, NSP can do admission control of IP QoS requests from end-users, and provide policing service of the admitted IP QoS based on the authorization information given at the signaling phase. (b) of Fig.1 shows an example scenario of Admission-based Direct

DiffServ solution to support dynamic QoS for simple SIP multimedia applications. Since UA is performing the role of QoS PEP(Policy Enforcement Point) and BRAS(Broadband Remote Access Server) of QoS-enabled DSL networks is acting as QoS PDP(Policy Decision Point), in the Admission-based Direct DiffServ, the standard PEP-PDP COPS(Common Open Policy Service) protocol [6] can be used for the QoS signaling between UA and BRAS. The PIB(Policy Information Base) defined as a named object for the UA-BRAS COPS protocol will convey identification information, QoS attribute values, and DiffServ classification information for each media stream of a multimedia application. Though the UA-BRAS COPS signaling protocol makes Admission-based Direct DiffServ solution more complex than Direct DiffServ solution, it can provide policing service of the admitted QoS for authorized users to protect from non-authorized users. Therefore, this solution will be very useful to support P2P SIP multimedia applications in most broadband access networks where QoS policing is necessarily required.

3) In Indirect DiffServ solution, some QoS proxy servers of a multimedia application will be responsible for supporting dynamic QoS on behalf of the end-user entities of the application. SIP default(inbound/outbound) proxy servers will become QoS proxy servers in QoS-enabled DSL access networks. COPS protocol can be also used for the QoS signaling between default proxy server and BRAS, same as in the UA-BRAS signaling of Admission-based Direct DiffSrev solution. (c) of Fig. 1 shows example scenario of Indirect DiffServ solution to support dynamic QoS for simple SIP multimedia applications. Default proxy servers, taking the role of QoS proxy servers, collect identification information and QoS attribute values for each media stream and determine DiffServ class, by capturing SDP offer and answer exchanged between UAC and UAS via INVITE and 180 RINGING messages. When the remote default proxy server receives 200 OK message indicating successful QoS negotiation from UAS, it delivers the identification information and QoS attribute values to its remote BRAS and requests IP QoS admission, by sending COPS REQ message. Remote BRAS admits the IP QoS request by sending COPS DEC message, after checking the configuration profile for the ASP and its resource allocation status. Admitted remote default proxy server sends 200 OK response to local default proxy server. Local default proxy server, after receiving 200 OK message, performs similar IP QoS admission procedure with its local BRAS by using COPS protocol, and delivers the 200 OK message to UAC, if successfully admitted. And then, BRAS will be ready to perform packet classification and DSCP marking for media stream incoming from the user, and aggregate queuing and prioritization.

In the Indirect DiffServ solution, BRAS is injection point of IP QoS and end-user systems are not involved in the IP QoS enforcement. This means that Indirect DiffServ can be easily deployed in the legacy QoS-unaware end-system environments. On the other hand, Indirect DiffServ solution can be supported only in ASP environments where there are QoS proxy servers and COPS signaling is supported between QoS proxy server and BRAS. Moreover, Indirect DiffServ has a significant disadvantage that there is no ways to support IP QoS in the access network ahead of BRAS because end-systems are not involved in the QoS support. Consequently Indirect DiffServ solution will be very useful to support dynamic QoS for ASP SIP multimedia applications in legacy broadband access networks where most end-systems are QoS-unaware.

4) Hybrid DiffServ solution is an integrated solution of Direct DiffServ and Indirect DiffServ solutions. Whereas end-systems of Indirect DiffServ solution are not involved in any activities to support IP QoS, Hybrid DiffServ solution additionally allows end-systems to directly perform IP QoS enforcement activities such as DSCP marking, based on the identification information, QoS attribute values, and DiffServ classification information for each media stream collected during QoS negotiation procedure. This will solve the problem of Indirect DiffServ solution that IP QoS is not supported in access network ahead of BRAS. (d) Fig. 1 shows an example scenario of Hybrid Diff-Serv solution to support dynamic QoS of simple SIP multimedia applications.

3 Concluding Remarks

In this paper, we proposed four different dynamic QoS solutions based on DiffServ. Direct DiffServ solution is useful to support P2P SIP multimedia applications in enterprise network environments where end users can be trusted. Admission-based Direct DiffServ can be easily applied to support P2P SIP multimedia applications in most broadband access networks where QoS policing is necessarily required. Indirect DiffServ solution will be effective in supporting dynamic QoS for ASP SIP multimedia applications in legacy broadband access networks where most end-systems are QoS-unaware, since end-user systems are not involved in the IP QoS enforcement. Hybrid DiffServ solves the problem of Indirect DiffServ solution that IP QoS is not supported in access network ahead of BRAS by additionally allowing end-systems to directly perform IP QoS enforcement activities such as DSCP marking.

References

1. Lakkakorpi, J., Strandberg, O., Salonen, J. : Adaptive Connection Admission Control for Differentiated Services Access Networks. IEEE JSAC, Vol. 23, NO. 10 (Oct. 2005)
2. Salsano, S., Veltri, L.: QoS Control by Means of COPS to Support SIP-Based Applications. IEEE Network (March/April 2002)
3. DSL-Forum TR-059 : DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services (Sept. 2003)
4. Rosenberg , J. et al. : SIP : Session Initiation Protocol. IETF RFC 3261(June 2002)
5. Rosenberg, J., Schulzrinne, H. : An Offer/Answer Model with Session Description Protocol(SDP). IETF RFC 3264 (June 2002)
6. Durham, D. et al. : The COPS(Common Open Policy Service) Protocol. RFC 2748 (Jan. 2000)

IP Traceback Algorithm for DoS/DDoS Attack*

Hong-bin Yim and Jae-il Jung

Department of Electrical and Computer Engineering Hanyang University,
17 Haengdang-dong, Sungdong-gu, Seoul, 133-791, Korea
hbyim@mnlab.hanyang.ac.kr, jijung@hanyang.ac.kr

Abstract. DoS(Denial of Service) / DDoS(Distributed Denial of Service) attacks threaten Internet security nowadays. However, the current Internet protocol and backbone network do not support traceback to know attacker's real location. Many methods to defend DoS/DDoS attack have been proposed. However these kinds of methods cause network overhead because they use many packets to reconstruct an attack path. In this paper we propose effective probability marking methods and a pushback algorithm to reduce network overhead.

1 Introduction

With commercialization of Internet, Internet security has become an increasingly important issue. DoS attacking programs create many forged packets and send these packets to one or many systems directly. In comparison with DoS, DDoS attacks occur along many attack paths by many attacking programs. It is difficult to know the real location of the attacker because this attack scenario uses a spoofed IP address [1]. There are many kinds of DoS/DDoS attack defense methods, however these kinds of methods cause network overhead because many packets are needed to reconstruct the attack path.

In this paper, we propose an effective traceback method. The rest of this paper is organized as follows. Section 2 outlines related works about IP traceback methods, section 3 outlines the effective IP traceback algorithms and section 4 shows the simulation results. Finally section 5 shows the conclusion and future work.

2 Related Works

There are three probabilistic packet marking methods such as node sampling, edge sampling and advanced packet marking[3,4]. The node sampling method requires many packets to reconstruct the attack path. The edge sampling method

* (This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment) (IITA-2005-(C1090-0502-0020))).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 558–561, 2006.
© Springer-Verlag Berlin Heidelberg 2006

marks the result of the XOR operation with its own IP address and the IP address of the previous router. This method is better than the node sampling method in reconstructing the attack path. Advanced packet marking method provides an authentication function during packet marking[6].

Multi-edge marking[5] appends the adjacent segment attack path to the record route IP option of the packet as it travels through the network from attacker to victim. Because of the limited space in the IP header, it can not mark all the routers' IP addresses into the record route IP option if the attack path is longer than 9. If there are no other IP options in use, the record route IP option can contain 9 IP addresses maximum.

3 Effective IP Traceback Algorithm

3.1 Modified IP header

At present, 8bits TOS(Type of Service) field the least 2 bits are not used[2]. Thus this paper uses these 2bits as a marking flag(MF) and a pushback flag(PF). These 2bits combination has 4 different pieces of information. Figure 1 shows the modified IP header that this paper proposes.

- MF/PF(1/0) : Excute packet marking algorithm.
- MF/PF(0/1) : Excute pushback algorithm.
- MF/PF(1/1) : Make an information packet.
- MF/PF(0/0) : Excute normal routing algorithm.

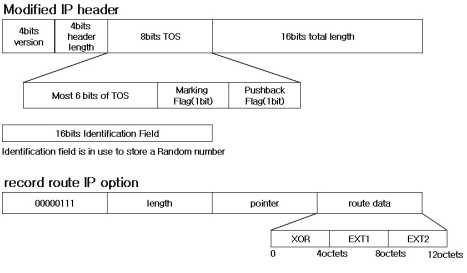


Fig. 1. Modified IP header and data format of record route IP option

When the edge router sets the marking flag to 1 with probability p, the record route IP option is opened. The route data field is used for the XOR, EXT1 and EXT2 fields. The XOR field stores the XOR value, the EXT1 and EXT2 fields store some information to traceback. Each router has a Route Information Table. This table consists of 2 field, packet ID and exclusive OR value. Packet ID field is in use to store 16bits random value to distinguish the XOR value of packet. Exclusive OR value field is in use to store XOR value that comes from the EXT2 field in the record route IP option of a packet.

Presently, the ratio of the fragmented packet of all Internet packets is less than 0.25%[7]. Thus the identification field and the fragmentation field are not

used to identify which packet is fragmented. This paper uses the identification field to store a 16bits random number to distinguish the XOR value of each packet. To use this field, most 3bits of the fragmentation field are set to 010. It means this packet is not fragmented. Most 3bits of the fragmentation field is named fFlag.

3.2 Probabilistic Single Packet Marking (PSPM), Simple Pushback (SP) and Simple Path Reconstruction (SPR) Algorithm

In PSPM algorithm, the edge router starts packet marking with probability p. The router that received the marking packet starts packet marking according to the packet marking algorithm. In SP algorithm, a victim that has received a marking packet will have their pushback flag set to 1. The victim's IP address is inserted into the EXT2 field in the IP header. The victim sends it to victim's edge router. When all routers that are received this packet starts pushback to find the attack path according to the pushback algorithm. Figure 2 shows full probabilistic single packet marking and simple pushback algorithm.

```
Packet Marking Algorithm:
    R is a router's own IP address
    for each packet w
            let u be a random number from [0,1)
            If u < q then
                    Let x be a 16bits random number
                    store x into w.Identification
                    set to 1 Marking Flag
                    store 010 into w.fFlag
                    store w.SourceAddress into w.EXT1
                    store w.EXT1 into w.EXT2
                    store w.EXT2 into w.XOR
            else if (MF= =1 and PF= =0) then
                    store w.EXT2 and x into RIT
                    store w.EXT1 into w.EXT2
                    store w.XOR into w.EXT1
                    store w.XOR^R into w.XOR
                            if victim receives this packet then
                                    victim sets PF to 1 and pushback algorithm is started
                            endif
                    endif
            endif

Signature ^ means exclusive ORing operation.

Pushback algorithm:
        R is a router's own IP address
        if (MF= =0 and PF= =1) then
                store R into w.SourceAddress
                store EXT1^RIT into w.DestinationAddress
                store RIT into w.EXT1
                        send this packet to upstream next hop router and make a new information
                        packet that R is a source address and EXT2 is a destination address, and send a
                        information packet to victim

                        if(Ext1= =RIT) then
                            stop pushback algorithm
                        endif
            endif

Signature ^ means exclusive ORing operation.
```

Fig. 2. Probabilistic Single Packet Marking and Simple Pushback Algorithm

In SPR algorithm, all routers that have received a pushback flag packet make an information packet that includes the IP address of its own router. This information packet is sent to the victim. This packet's source IP address of IP header is the router's IP address and the destination IP address is the IP address of the victim. This information packet sets the marking flag and pushback flag to 1. This packet is transmitted according to the normal routing algorithm. The victim uses this packet to reconstruct the attack path. The initial TTL(Time To Live) value of a normal packet is set to 255. The victim can know the number of hops to the router using decrease of TTL value. The victim line up ascending order of decrease of TTL value, is the order of the attack path from attacker to victim. The ordering algorithm is simply the ascending order algorithm.

4 Simulation

In the simulation, we compare the following three traceback algorithms: the edge sampling method, the multi-edge method and the proposed traceback method.

In the case of a one attacker attack the number of packets in the proposed traceback scheme is decreased by an average of 16.5% compared to the multi-edge method. In the case of a 10 attackers attack the number of packets in the proposed traceback scheme is decreased by an average of 37.6% compared to the multi-edge method. This result shows that the proposed scheme is a more effective method than the multi-edge method in the case of DDoS attack.

Table 1. Required packet to reconstruct attack path when p=0.1

number of attackers	One attacker			Ten attackers		
number of hops	10	15	20	10	15	20
edge sampling	74	126	235	595	1344	2218
multi-edge	30	35	37	322	397	649
proposed scheme	23	28	35	191	292	362

5 Conclusion

In this paper, we have proposed an effective packet marking algorithm to reduce packets. This algorithm can be implemented in a router and uses record route IP option and unused fields in the IP header. The marking algorithm uses a probabilistic marking scheme and XOR operation to calculate IP addresses to reduce the IP header packet size.

According to the simulation result, the proposed traceback algorithm is more effective on a DDoS attack than the schemes available up to now. In the future, we also need to test on a large scale network to check performance of this algorithm in terms of the packet collection time.

References

1. Henry C.J. Lee, Vrizlynn L.L. Thing, Yi Xu, and Miao Ma, "ICMP Traceback with Cumulative Path, an Efficient Solution for IP Traceback", LNCS 2836, pp.124-135, 2003
2. "Internet Protocol : DARPA INTERNET PROGRAM PROTOCOL SPECIFICATIO", RFC791
3. K. Park and H. Lee. "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack", In Proc. IEEE INFOCOM 2001, page 338-347,2001
4. D. X. Song, A. Perrig, "Advanced and Authenticated Marking Scheme for IP Traceback", Proc, infocom, vol.2, pp.878-886, 2001
5. Chen Kai, Hu Xiaoxin, Hao Ruibing, "DDoS Scouter : A Simple IP Traceback Scheme" http://blrc.edu.cn/blrcweb/publication/kc1.pdf
6. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, "Practical Network Support for IP Traceback", SIGCOMM, 2000
7. Ion Stoica, Hui Zhang, "Providing Guaranteed Services Without Per Flow Management", SIGCOMM, 1999

An Open Service Platform at Network Edge

Dong-Hui Kim and Jae-Oh Lee

Information Telecommunication Lab.,
Dept. of Electrical and Electronics, Korea University of Technology and Education, Korea
{dhkim, jolee}@kut.ac.kr

Abstract. The last few years, users want various services that fit their needs and preferences. Many services are provided at network edges and are increasing in number. The open framework is needed for efficient service management and uniform deployment of personal services at network edges. In this paper, we propose the open service framework, which can be implemented as a platform for the personal service at network edges by using Open Pluggable Edge Service (OPES) concept. The proposed framework is composed of Databases, Policy Repository, Rule Manager, Policy Adapter, Admission Manager, Operations Support System (OSS) Adapter and OPES-based Service Delivery Manager for the deployment of personalized and QoS guaranteed services in a standard way. In order to perform the feasibility of this platform, we have implemented a simple example and shown some its results.

1 Introduction

Personalized service is to adapt the provisioned services to fit the needs and preferences of a user or a group of users in a static or dynamic way. The process of personalized service gathers information of user, device and service through their interactions and in turns, stores the collected raw information into database.

The edge service means service for deploying service by 3rd-party Service Developers and Network Provider at network edge. Because each edge has many services, it needs the method to manage and deploy services. In [1], a platform is proposed for managing and deploying various services, but it does not mention on the concrete relationship with personalized services.

So we propose the open service framework, which can be represented as the personalized service platform with the network and service control functionality for the dynamic provision of a variety of edge-based services, being deployed and managed.

Section 2 introduces the Open Pluggable Edge Service (OPES) architecture to be applied in the framework. Section 3 describes the proposed framework for deploying the personalized services in an effective way and section 4 implements a case of personalization service for the feasibility of the proposed framework. Section 5 provides conclusive remarks and directions for future work.

2 Open Pluggable Edge Service (OPES) Architecture

The OPES Working Group of Internet Engineering Task Force (IETF) has developed an architectural framework to authorize, invoke, and trace such application-level

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 562 – 565, 2006.
© Springer-Verlag Berlin Heidelberg 2006

services for HTTP. The framework is a one-party consent model in which each service is authorized explicitly by at least one of the application-layer endpoints.

The OPES, existing in the network edge, provides services that modify requests, modify responses and create responses. The architecture of OPES can be described by OPES entities, OPES flows and OPES rules. An OPES entity residing inside OPES processors is an application that operates on a data flow between a data provider application and a data consumer application.

An OPES entity consists of an OPES service application and a data dispatcher. An OPES service application can analyze and transform messages on data stream. A data dispatcher invokes an OPES service application according to an OPES rule-set and application-specific knowledge. If there is no needed service application in local OPES processor or is useful for OPES processor to distribute the responsibility of service execution in an OPES service application of remote Callout Servers, the data dispatcher invokes remote service application by communicating one or more Callout server(s). In this case, OPES Callout Protocol (OCP) is used for communication between a data dispatcher and a Callout Server.

OPES flows are data flows among a data provider, a data consumer and one or more OPES processor(s) in which a data dispatcher must be existed. The exchanges of data between a data provider and a data consumer are independent of protocols. For the simplicity and implementation of proposed platform, we select HTTP, a basic protocol of Web Services, as the example for the underlying protocol in OPES flows.

The rule-set is a superset of all OPES rules. The OPES rules consist of a set of conditions and their related actions which can be specified as when and how to execute OPES services on data stream. The data dispatcher examines the rules and invokes service application for offering the service at the points identified by the numbers 1 through 4.

3 The Open Service Framework

From the perspective of network topology, an access network for the user to access service might be constructed as access node, access switch and edge router. We present the open service framework, which can supply the personal services with the negotiated QoS requirements to be deployed using OPES concept at network edge.

The proposed open service framework is composed of Databases for Profiles, Policy Repository (PR), Rule Manager (RM), Policy Adapter (PA), Admission Manager (AM), OSS Adapter (OSSA) and OPES-based Service Delivery Manager (OSDM) as depicted in Fig. 1. Databases contain profiles of user, device and service to be used as basic information for the accomplishment of personalized services with their QoS requirements. The RM can perform the query operations to all of profiles. It sends service id, user id and qualified Service Level Agreement (SLA) for the user to the PR. And then the RM generates the determined rules by using collected profiles from Databases and received polices from the PA. The PA decides the suitable policies based on SLA for the service and network requirements. Then it sends the determined

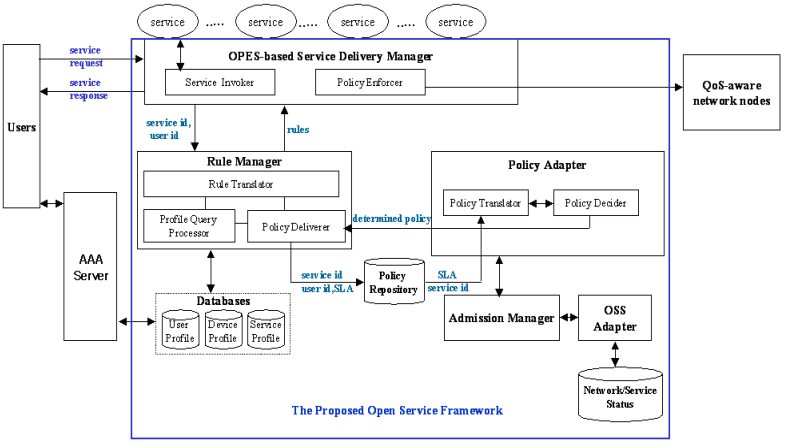


Fig. 1. The Open Service Framework

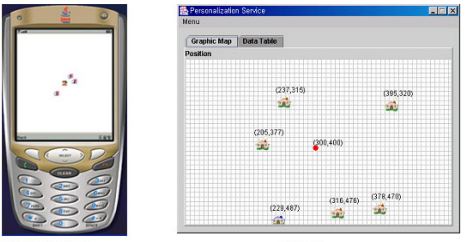
policies with their priorities to the RM. The AM can perform the functionality of whether the requested application service can satisfy the negotiated QoS requirements. In order to perform this functionality, the AM can access the OSSA, which offers the network and service management information for representing the status of network resource and service provision. Therefore, the OSDM can deploy many suitable application services to which the received rules from the RM are applied. Also, it enforces the established policies related to application services into QoS-aware network nodes such as edge router, access node and access switch.

4 An Implementation Case of Personalized Service

We have implemented a simple example by using the proposed open framework for the mobile device. We use Red Hat 9.0 for Server, MySQL for Database, Java Wireless Toolkit for client and XML technology (i.e., Java WSDL 2.0) for the processing of rules.

We suppose that the user wants to know about the nearest restaurants. The service priority of the restaurant searching is determined from the characteristics of its distance, preference and similarity in that order. We don't consider geographical features such as mountain, river and so on.

In Fig.2, (a) shows the result for cellular phone and (b) shows the result for laptop for searching the restaurants based on user preference. As depicted in Fig. 2, the result of adapted and personalized service can be displayed in different shapes according to user devices. They graphically show the nearest restaurants in accordance with the reflection of user preference. The user receives the information on the closest and the most preferable restaurants. Also the user might acknowledge the location information of found restaurants for their reservation or their more detailed information. If the user wants to know the restaurant information such as name, menu and distance, one just selects one's wanted restaurant from the display of cellular phone and laptop.



a) cellular phone b) laptop

Fig. 2. The Most Preferable Restaurants Found by the Personalization Service

5 Conclusion

We propose the open service framework, which can be represented as the personalized service platform with the network and service control functionality for the dynamic provision of a variety of edge-based services, being deployed and managed. In order to make this framework, we design the system components such as Databases, Policy Repository, Rule Manager, Policy Adapter, Admission Manager, OSS Adapter and OPES-based Service Delivery Manager.

However there are many Service Providers at the network edge and they want to provide service at other Network Provider edge. Network Provider requires Service Providers to comply with its API for deploying their services. For following the convenience and standard to Service Provider, we will extend the proposed framework towards Service Delivery Platform (SDP).

References

1. Falchuk, B., Chiang, J., Hafid, A., Cheng, Y.-H., Natarajan, N., Lin, F.J., Cheng, H.: An open service platform for deploying and managing services at network edges, Open Architectures and Network Programming, pp. 77 – 86, April 2003.
2. OPES Working Group: http://www.ietf.org/html.charters/opes-charter.html.
3. Dinesh C. Verma: Policy-Based Networking-Architecture and Algorithms, New Riders, 2000.
4. Timo Laakko, Tapio Hiltunen: Adapting Web Content to Mobile User Agents, IEEE Internet Computing, Vol 9. No 2, pp. 46-85, 2005.
5. Panayiotou, C., Andreou, M., Samaras, G., Pitsillides, A.: Time based personalization for the moving user, Mobile Business, pp. 128 – 136, July 2005.
6. Dong-Jun Lan, Chun Ying, Jun Liu, Wei Lu : Policy Lifecycle and Policy Enabled Telecom Open Service Delivery Platform, IEEE International Conference on Web Service, 2005.
7. Sun Microsystems.Inc: http://java.sun.com/products/sjwtoolkit/index.html.
8. Deitel: Java Web Service for Experienced Programmers, Prentice Hall , 2002.

Hybrid Inference Architecture and Model for Self-healing System*

Giljong Yoo, Jeongmin Park, and Eunseok Lee**

School of Information and Communication Engineering, Sungkyunkwan University
300 Chunchun Jangahn Suwon, 400-746, Korea
{gjyoo, jmpark, eslee}@ece.skku.ac.kr

Abstract. Distributed computing systems are continuously increasing in complexity and cost of managing, and system management tasks require significantly higher levels of autonomic management. In distributed computing, system management is changing from a conventional central administration, to autonomic computing. However, most existing research focuses on healing after a problem has already occurred. In order to solve this problem, an inference model is required to recognize operating environments and predict error occurrence. In this paper, we proposed a hybrid inference model – ID3, Fuzzy Logic, FNN and Bayesian Network – through four algorithms supporting self-healing in autonomic computing. This inference model adopts a selective healing model, according to system situations for self-diagnosing and prediction of problems using four algorithms. Therefore, correction of error prediction becomes possible. In this paper, a hybrid inference model is adopted to evaluate the proposed model in a self-healing system. In addition, inference is compared with existing research and the effectiveness is demonstrated by experiment.

1 Introduction

A computer system would satisfy the requirements of autonomic computing, if the system can configure and reconfigure itself by knowing the operating environments, protect and heal itself from various failures or malfunctions. The core of *autonomic computing*, a recently proposed initiative towards next-generation IT-systems capable of 'self-healing', is the ability to analyze a data in real-time and to predict potential problems [1]. Currently, most self-healing systems perform healing after error occurrence [2]. Sun has developed a new architecture for building and deploying systems and services capable of Predictive Self-Healing [3]. However, this is predicted through limited elements, because of healing in the same manner as IBM (vender-dependant). In order to know the environments and detect failure, an autonomic system needs the capability of acquiring the information through self-monitoring.

* This work was supported in parts by *Ubiquitous Autonomic Computing and Network Project*, 21th Century Frontier R&D Program, MIC, Korea, ITRC IITA-2005-(C1090-0501-0019), Grant No. R01-2006-000-10954-0, *Basic Research Program* of the Science & Engineering Foundation, and the *Post-BK21 Project*.

** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 566 – 569, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, a hybrid inference model is proposed to solve this problem. These proactive prediction and probing capabilities will provide the system management components with the pertinent information such that self-healing are possible for critical system resources. These models adopt a selective model, depending on the system situation, for self-diagnosing and prediction of problems. In this paper characteristics of each algorithm are detailed through a comparison of four inference model and a demonstration by experiment.

The remainder of this study is organized as follows: Section 2 describes the proposed architecture and model; Section 3 describes an evaluation through its implementation and experiments; and Section 4 presents the conclusion.

2 Hybrid Inference Method for Self-healing

2.1 A Proposed Architecture and Model

Using inference algorithms on self-healing system results is described by the ID3, Fuzzy logic, FNN and Bayesian Network.

Inference models can be chosen according to situation analysis. Therefore, a self healing system that can efficiently heal and manage the system is designed and implemented. This paper proposes architecture for efficient use of the four inference model algorithms in the self-healing system. The architecture consists of five modules, *System monitoring Module*, *Resource Level Evaluator*, *Prediction Model Selector*, *Prediction Model Algorithm Executor* and *Model Updater*. These modules is contained a partial function in Monitoring Agent [4].

We used four algorithms for hybrid prediction about system's situation.

Firstly, ID3 builds a decision tree from a fixed set of historic data. The resulting tree is used to classify future system situation. This algorithm can obtain the fastest and more effective result using the four algorithms when the early historic data is rich. The ID3 is used in the following situation.

- The early stage, in case that data to make Decision tree is much
- In the case that system is needed quick prediction like bank's web server

Secondly, Fuzzy logic can be a dangerous method that applies fuzzy in an actual system. But, ambiguousness can be useful because the exact boundary of error can not be defined in real world. Fuzzy logic has a four level, that is, Fuzzy Matching, Inference, Combining Fuzzy Conclusion, and Defuzzification [5].

Thirdly, the fuzzy neural network is used in the following situation.

- In the case that the self-learning is required by the system.
- In the case that a developer can't intervene a system.
- In the case that the number of rules is not enough to understand the status of the system

Fourthly, Bayesian networks attempt to represent expert knowledge in domains where expert knowledge is uncertain, ambiguous, and/or incomplete [6]. Bayesian networks are based on probability theory. This paper uses the casual inference and the diagnostic inference technique, using various inference methods. Bayesian network is used in the following situation.

- In the case that reliability about all situation of system is required.
- In the case that the target system analyzes the cause from a result of system situation.

2.2 Applying TMR Method

Triple Modular Redundancy (TMR) [7] is method used for increasing fault tolerance in safety critical software system. When four prediction algorithms can't select a suitable algorithm in specified situation, the *Monitoring Agent* predicts the system situation by the comparison among each algorithm using TMR. TMR is built up from three equal components (all three components should produce the same output for the same input) and an output selector. Four algorithms is flexibly changed and replaced according to the system situation. TMR assumes that the algorithms must have equal input and output. If the one algorithm has different output, it is recognized as a fault. And the others are recognized as a correct value of inference.

3 Implementation and Evaluation

In order to evaluate hybrid inference model's performance, proposed in this paper, firstly, a prototype of a self-healing system is implemented, secondly, this prototype is applied to a self-healing system, thirdly, it is compared with the ID3, and finally, specific characterizes are compared by an experiment.

In the experiment, the operating system used was Windows XP professional and the CPU was an Intel Pentium 3GHz, with 512Gbyte's of RAM and network capacity of 100Mbps. The self-healing system was installed [4], and the efficiency of the hybrid algorithm was evaluated.

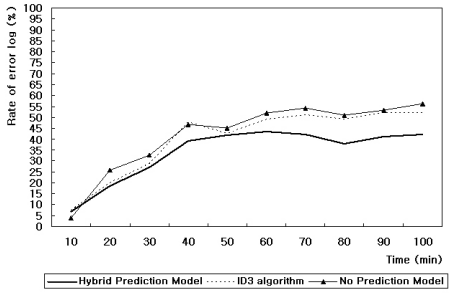


Fig. 1. Error rate changed by the time flow

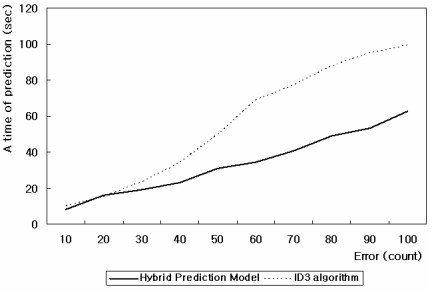


Fig. 2. A processing time to predict the system situation by the number of error

In the first experiment, comparison of the existing system was performed, that only used the ID3 to predict the proposed system. As a result, it was concluded that a rate of error log is reduced through use of the suitable inference model according to system requirements. The above experiment result is presented in Figure 1.

The time taken for prediction in the second experiment was measured in Figure 2. First, we assume a learned data has an enough quantity. It proposed model's efficiency was proven through comparison of the number of errors occurring in the system. If the number of error is fewer, ID3's result displays similar performance such as the result of applying a hybrid model. However, the number of errors was high, and the prediction time of the hybrid model, which predicts the characteristic of the system as well as resource information, was shorter.

4 Conclusion

Previous monitoring technologies of the self-healing system dissatisfied a requirement of the ubiquitous computing. In recent years, numerous studies have attempted to find and explore self-healing systems, such as IBM [2], and Sun [3]. However, it is a healing of vender-dependent and the use of a single inference model. This paper proposed a hybrid inference model to recognize operating environments and predict error occurrence. Therefore, we can perform an efficient prediction by the system's status and environment through a hybridization of inference model. Four algorithms proposed in this paper are clearly divided with characteristics and can be used in hybrid inference models, according to the system request. We designed architecture to support hybrid inference model for self-healing system. We compare an efficiency of algorithm by experiment about the time of prediction, the correctness of prediction and system's load. So, we made up for the weak point in our previous work [4].

References

1. R.K. Sahoo, A. J. Oliner, I.Rish, M. Gupta, J.E. Moreira, S. Ma, "Critical Event Prediction for Proactive Management in Large-scale Computer Clusters", ninth ACM SIGKKD international conference on Knowledge discovery and data mining, pp. 426-435, 2003
2. B. Topol, D. Ogle, D. Pierson, J. Thoensen, J. Sweitzer, M. Chow, M. A. Hoffmann, P. Durham, R. Telford, S. Sheth, T. Studwell, "Automating problem determination: A first step toward self-healing computing system", IBM white paper, Oct. 2003
3. Sun Microsystems: Predictive Self-Healing in the Solaris 10 Operating System, http://www.sun.com/bigadmin/content/selfheal
4. Jeongmin Park, Giljong Yoo and Eunseok Lee, "Proactive Self-Healing System based on Multi-Agent Technologies", ACIS International Conference on Software Engineering Research, Management & Application(SERA 2005), IEEE, pp.256-263, Aug.2005
5. Kwang H.Lee, 'First Course on Fuzzy Theory and Applications', Advances in Soft Computing, Springer, 2005
6. Sucheta Nadkarni, Prakash P. Shenoy, "A causal mapping approach to constructing Bayesian networks", Decision Support Systems, Vol.38, pp.259-281, Nov.2004
7. J. Von Neumann, 'Probabilistic logics and synthesis of reliable organisms from unreliable components' in Automata Studies, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, pp. 43–98, 1956

A Node Management Tool for Dynamic Reconfiguration of Application Modules in Sensor Networks*

Sunwoo Jung, Jaehyun Choi, Dongkyu Kim, and Kiwon Chong

Department of Computing, Graduate School, Soongsil University, Seoul, Korea
{sunoo7906, uniker80, mapssosa, chong}@ssu.ac.kr

Abstract. The configuration management of a sensor network composed of tens or up to thousands of nodes is difficult due to the limit of memory space and energy consumption. This paper presents an approach for dynamic reconfiguration of application modules in sensor networks based on Nano-Qplus. The version information from proposed methods is registered into NVSync (Nano-Qplus Version Synchronization), a node management tool. It dynamically reconfigures application modules on the nodes by synchronizing them with the node information map on the centralized repository. Our approach utilizes efficient usage of limited memory by systematically managing application modules of each node composed of sensor network. Additionally, it reduces unnecessary energy consumption by dynamic reconfiguration of application modules on the nodes. Therefore, it can flexibly deal with change of application environment of the node.

1 Introduction

Researchers have done much work on wireless networks for a number of years and have developed fairly sophisticated sensor network systems in the face of strict constraints such as low power, low cost, small size, fault tolerance, flexibility, and security. The reconfiguration of application modules for sensor networks is challenging about above constraints at each node. Reconfiguration and self-adaptation are important factors of sensor networks that are required to operate in dynamic reconfiguration that includes functional changes and nonfunctional performance improvements.

Dynamically adaptive application comprises tasks that detect internal and external changes to the system, reflecting the new application environment conditions. Wireless sensor networks, in particular, require reconfigurable capabilities that enable them to handle a multitude of nodes. Consequently, existing systems that collect and aggregate data in sensor networks provide an active tool for managing each node.

Our approach applies the domain of robust, fault-tolerant embedded systems. It is able to reconfigure various changes tailored for application of nodes. The system can quickly find a new configuration and adapt to environment changes. This paper proposed a systematic node management tool of node application modules using the minimum memory resources in the sensor nodes. Based on this approach, the tool uses bug modifications, function updates, and application changes on the nodes. This tool is implemented based on Nano-Qplus[6], developed by ETRI, Korea. If we utilize the proposed tool, therefore, developers develop application modules of the nodes simultaneously and we can flexibly deal with changes on the node.

* This work was supported by the Soongsil University Research Fund.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 570 – 573, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Works

The sensor network is defined that it can collect the necessary sensing information and each node has a processor that can process collected information and wireless telecommunication device that can transmit it. Typical sensor networks consist of tens, if not hundreds, of nodes, including special sink nodes that connect them to global networks, such as the Internet. Communication occurs regularly over multiple hops, and due to frequent poor link quality, reliable data collection at the sink node is a significant problem. Namely, the sensor networks will provide the bridge between the sensors and the physical world due to their ability to observing and controlling the sensing data in real-time. The above described features ensure a wide range of applications for sensor networks[1]. There were some researches in relation to sensor network operating systems such as TinyOS[2], SOS[3], Maté [4], MANTIS[5] and Nano-Qplus[6]. Especially, Nano-Qplus, our approach target OS, supports ultra-small kernel size(smaller than 10KB), distributed, real-time, and smart operating system that is suitable for various application areas such as medical care, environment, disaster prevention, digital home, national defense, and industrial equipment.

Prevalent sensor network operating systems have some problems as follows: (1) use only version management system tools by targeting simultaneous development, (2) do not support automatic version synchronization between development environment nodes and real nodes, (3) rarely provide the fine-grained modular update to fit the given memory limit or to optimize the system.

3 System Architecture

We have prototyped an architecture for software reconfiguration shown in Figure 1. It was a better way to update a sensor network using a centralized database. It is possible to increase the reusability, self-adaptability, and flexibility. However, we prototyped our approach with the current limitation of the Nano-Qplus such as high update cost caused by reconfiguring a full application image and testing in a smaller scale sensor networks. We have plans to extend the modular update referred to SOS towards the modular-reconfigurable Nano-Qplus environment and to verify effectiveness and constraints such as energy consumption and memory limitation.

The NVSync(Nano-Qplus Version Synchronization), a node management tool primarily requires loading of a prerequisite image on an initial target node through the serial line and then gathers the node version information on the node and that on the repository. It updates the version information stored in the node information map for each node and reconfigures the application image using the target node information, when the role of the target node is changed. Most of the energies are wasted when it gathers the sensing data in the sensor field. The application module of a sensor node can be modified to gather proper data using the proposed approach with energy efficiency, when the sensor field is changed. Moreover it is possible to modify the application modules of a node without removing the existing node, when the application

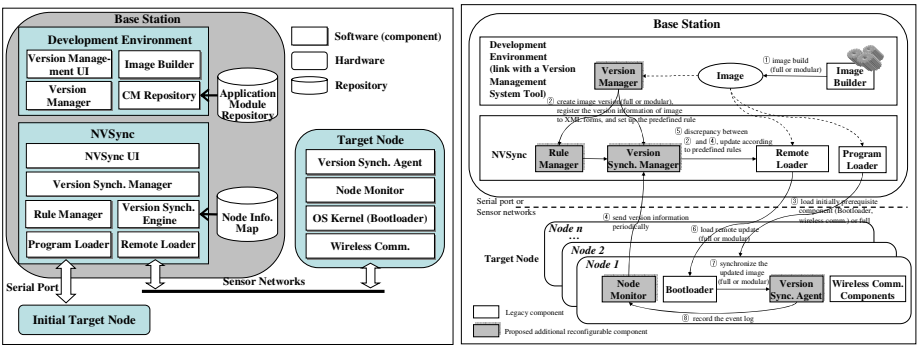


Fig. 1. System architecture(left fig.) and system behavioral sequence (right fig.) for the NVSync

Table 1. The components for software reconfiguration

Component	Descriptions
NVSync UI	▪ Displays entire user interface of SNM.
Version Synchronization Manager	▪ Registers/changes new or modified image's node information in the node information map from the Version Manager. ▪ Maps/remaps the node information map and the sensor network nodes. ▪ Executes the version reconfiguration after comparing the node version information of the Version Manager and the node information of the Node Monitor on the node. ▪ Applies the predefined rules of each node from the Rule Manager to reconfiguration.
Rule manager	▪ Selects the desired reconfiguration node. ▪ Sets up the predefined rule consists of 3 types – user mode, periodic mode, and direct mode.
Version Synchronization Engine	▪ A centralized database for the node information map.
Program Loader	▪ Primarily load initial application image built from Image Builder in the development environment.
Remote Loader	▪ Loads a modified application image through the wireless networks.
Node Monitor	▪ Stores version information of each node and periodically sends it to the Version Synchronization Manager of the NVSync.
Bootloader	▪ consists reconfigurable environment of application images and ports updated images through the Wireless Communication Components.
Version Synchronization Agent	▪ synchronizes updated image and records the event logs showing whether update is update complete or incomplete.

module of a node is changed. Because the information is stored in a node information map, it is easy to check the status of each node.

For instance, there might be a *sensornode1* that sense a gas leak at the gas depository1. In case the gas depository1 is moved to gas depository2, data gathered by the node is not necessary any more. If the node gathers unnecessary data, the energy of the node is wasted when the node sends the data to a sink node. In such a case, the developer generates a *sensornode1* image to discontinue unnecessary tasks in the old depository and a *sensornode7* image to sense a gas leak in the new depository using the Image Builder. The Remote Loader of the NVSync sends the Bootloader on each node two generated image. The image is dynamically reconfigured on the node by the Bootloader in a user mode. Consequently, each node can collect necessary data without wasting unnecessary energy.

4 Conclusion and Future Works

This paper suggests version synchronization approach for improving visibility changes of application modules of nodes and reconfiguring application through the NVSync. Each node has many constraints such as the lack of memory space and energy consumption. We presented an approach for constraints-considered software reconfiguration in sensor networks. The reconfiguration process takes place in a base station that can communicate to all the sensor nodes. This enables developers to develop application modules simultaneously and enhances the visibility of application version changes. The main contribution of our approach utilizes efficient usage of limited memory by managing systematically application module of each node composing sensor network and reduces unnecessary energy consumption by automatic and dynamic reconfiguration of application modules on the nodes. Therefore, this can flexibly deal with changes of application environment of the node. This approach, while relatively sophisticated, does not seem to pose theoretical challenges. However there are additional challenges for modularizing about application and supporting dynamic reconfiguration environments. Anyway applications must be reconfigured flexibly and optimally.

In the future, we will verify efficiency of this approach on energy consumption and memory use using the *MATLAB* simulation tool. Although the reconfiguration is achieved manually in our testing implementation, we have demonstrated its advantages. In addition, we will complete the implement of Nano-Qplus modular reconfiguration. Robustness of the reconfiguration method is also a significant challenge. The drawback of our approach, the necessity for modular-switching functions, arises because of the full-configured modules replacement nature of Nano-Qplus. Currently, we investigate these issues by implementing our approach using SOS[3].

References

1. Ian F. Akyildiz, Weilian Su, Yogosh Sankarasubramaniam, and Erdal Cayirci, “A Survey on Sensor Networks,” IEEE Communications Magazine, pp.102~114, Aug. 2002
2. Hill. J., Szewczyk. R., Woo. A., Hollar. S., Culler. D., and Pister. K. “System architecture directions for networked sensors,” In Proceedings of the ninth international conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), ACM Press, pp.93-104
3. Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler and Mani Srivastava, “A Dynamic Operating System for Sensor Nodes,” In proceedings of 3rd International Conference on Mobile Systems, Applications, and Services(MobiSys 2005) on the USENIX, pp.163-176, 2005.6
4. Levis. P., and Culler. D., “Maté: A tiny virtual machine for sensor networks,” In International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA (Oct. 2002).
5. Abrach. H., Bhatti. S., Carlson. J., Dai. H., Rose. J., Sheth. A., Shucker. B., Deng. J., and Han. R., “MANTIS: system support for multimodal networks of in-situ sensors,” In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (2003), ACM Press, pp. 50-59
6. Introduction and download of Nano-Qplus, http://www.qplus.or.kr

Path Hopping Based on Reverse AODV for Security

Elmurod Talipov, Donxue Jin, Jaeyoun Jung, Ilkhyu Ha,
YoungJun Choi, and Chonggun Kim*

Department of Computer Engineering,
Yeungnam University, Korea
elmurod@ynu.ac.kr, donghak@yumail.ac.kr,
e-mail@yumail.ac.kr, ilkyuha@yumail.ac.kr,
yjchoi@yu.ac.kr, cgkim@yu.ac.kr

Abstract. In Ad hoc networks, malicious nodes can enter in radio transmission range on the routing path and disrupt network activity. Therefore, protecting from intrusion of malicious node and enhance data security is an important issue on Ad hoc networks. In this study, we provide a path hopping method based on reverse AODV (R-AODV). By reverse AODV, source node builds multipath to destination and adaptively hops available paths for data communications. Hopping paths can protect data from the intrusion of malicious nodes. We propose an analytic method to expect intrusion rate and a path hopping routing mechanism to implement simulation models using NS-2.

Keywords: Path-hopping, reverse AODV, Ad hoc network security.

1 Introduction

A mobile ad hoc network is a dynamically self-organizing network without any central administrator or infrastructure support. If two nodes are not within the transmission range of each other, other nodes are needed to serve as intermediate routers for the communication between the two nodes [1, 2].

In ad hoc wireless networks, transmitted data is susceptible to potential attacks. Eavesdroppers can access secret information, violating network confidentiality. Hackers can directly attack the network to drop data packets, inject erroneous messages, or impersonate as a member node. To increase security, physical protection of the network from malicious node is important.

In this study we propose path hopping based on reverse AODV [2]. In R-AODV, which is an easy multipath searching method, destination node uses reverse RREQ to find source node rather than a unicast reply. It reduces path fail correction messages and also source node builds partial or complete non-disjoint multipath from source to destination. Hopping paths means source node sends each data packet through different paths each time, therefore eavesdropper will not get whole data and also its intrusion to network become harder [3-7].

Physical protection of data from malicious invader is an important security method. It can decrease or prevent packet loss by active malicious nodes [7].

* Correspondence author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 574 – 577, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Path Hopping Method on Multipaths

Path hopping based on reverse AODV routing protocol (PHR-AODV) is presented. Since PHR-AODV is reactive routing protocol, no permanent routes are stored in nodes. The source node initiates route discovery procedure by broadcasting the RREQ message. When destination node receives first route request message, it generates so called reverse request (R-RREQ) message and broadcasts it. By receiving R-RREQ messages source node simply builds partial non-disjoint multipath and hops one by one while sending data packets [2]. In PHR-AODV the number of paths from one source to destination is decided as the number of edges from source node.

Purpose of our study is to strength security of routing and decrease possible intrusions of malicious nodes. PHR-AODV maintains list of routing paths from source to destination. The messages are sent by multipaths. The order of path selection can be variable. We just accept sequential order in this study. During the communication, a path failed then the path is eliminated from the list. When no path is remained in the list, the source node sends RREQ for establishing new multipaths.

Path disjointing is an important point of multipath routing. Several disjointing paths are studied, such as [2], [5] and [7]. Generally multipath can be classified in three groups: node-disjoint (complete disjoint), link disjoint, non-disjoint [3-5]. PHR-AODV builds compete or partial node-disjoint depend on topology. Figure 2 shows partial non-disjoint case.

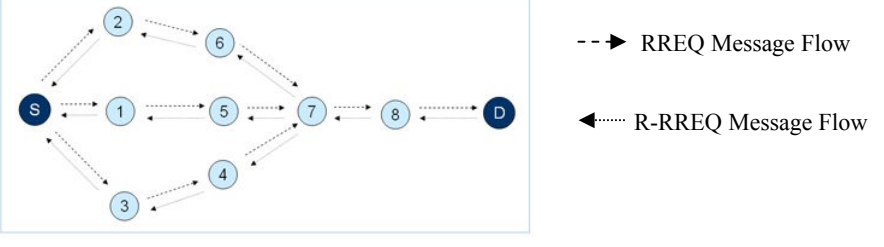


Fig. 2. Example of partial non-disjoint

Complete node-disjoint multipath is good for security, but partial node-disjoint multipath is also effective for security. We provide analytic method to estimate security. Let's assume some parameters: N_p is he number of nodes in routing path, N_{all} is the number of all nodes in network, M is the number of malicious nodes, (we assume that only one malicious node exists in the network), S is the number of paths from a source to a destination, ρ_m is probability of active malicious nodes.

$$\rho_m = \left(N_p \bullet M\right) / N_{all}. \tag{1}$$

We can calculate ρ_i, malicious node intrusion rate, as follows

$$\rho_i = \rho_m / S . \tag{2}$$

From formula 2, we obtain figure 3. The figure shows that increasing the number of paths derives decreasing of intrusion rate by a malicious node.

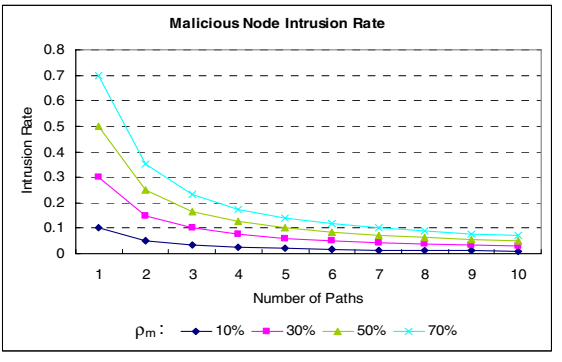


Fig. 3. Intrusion rate by a malicious node

3 Performance Results

We describe the simulation environment used in our study and then discuss the results in detail. Our simulations are implemented in Network Simulator (NS-2) [6].

We compare performance of AODV, R-AODV and PHR-AODV. Figure 4 shows packet deliver ratio of each protocol. R-AODV has better delivery ratio than other protocols have. PHR-AODV delivery ratio is less than other protocols, because it maintains more paths than others. Figure 5 shows the control packet overhead for setting routing paths. PHR-AODV has less packet overhead than that of R-AODV.

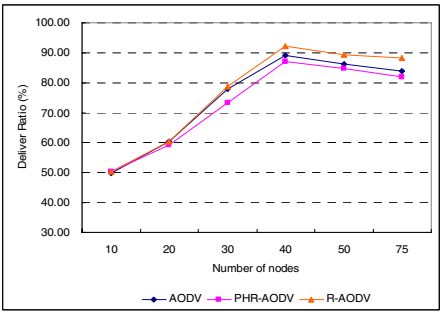


Fig. 4. Packet Delivery Ratio, when the number of nodes varies

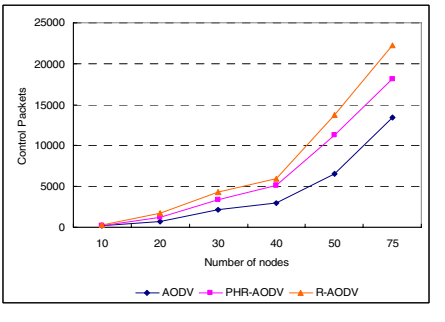


Fig. 5. Control Packet Overhead, when number of nodes varies

Figure 6 shows the average remained energy of each protocol. Figure 7 shows energy difference to express the distribution rate. PHR-AODV has less energy difference and balanced energy than others.

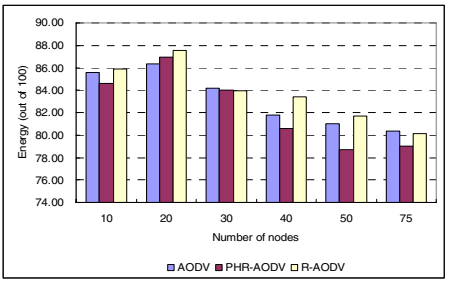


Fig. 6. Average energy remained, when number of nodes varies

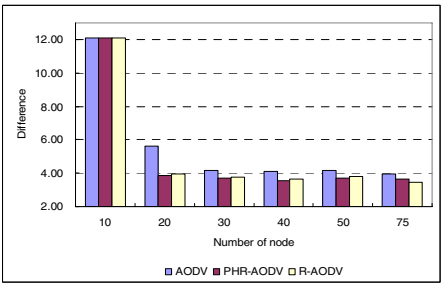


Fig. 7. Energy Difference, when number of nodes varies

4 Conclusions

Security is a significant issue in ad hoc networks. Intrusion of malicious nodes may cause serious impairment to the security. To decrease effect of malicious nodes, we proposed the idea of path hopping based on reverse AODV, in which the source node attempts to hop among available paths and split data. We conducted extensive analytic model and a simulation study to evaluate the performance of PHR-AODV with the R-AODV and AODV using NS-2. The results show that PHR-AODV maintains reasonable packet delivery ratio, energy consumption and energy distribution while increasing security of network. Our future work will focus on studying practical design and implementation for PHR-AODV.

References

1. C. Perkins, E. Belding-Royer Ad hoc on-Demand Distance Vector (AODV) Routing, RFC 3561, July 2003
2. Chonggun Kim, Elmurod Talipov, and Byoungchul Ahn, "A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks" , LNCS 4097, pp. 522 – 531, 2006.
3. C. K.-L. Lee, X.-H. Lin, and Y.-K. Kwok, "A Multipath Ad Hoc Routing Approach to Combat Wireless Link Insecurity," Proc. ICC 2003, vol. 1, pp. 448–452, May 2003.
4. S.-J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," Proc. ICC 2001, vol. 10, pp. 3201–3205, June 2001.
5. M. K. Marina and S. R. Das "On-Demand Multi Path Distance Vector Routing in Ad Hoc Networks," Proc. ICNP 2001, pp. 14– 23, Nov. 2001.
6. NS, The UCB/LBNL/VINT Network Simulator (NS), http://www.isi.edu/nsnam/ns/, 2004.
7. Zhi Li and Yu-Kwong Kwok, "A New Multipath Routing Approach to Enhancing TCP Security in Ad Hoc Wireless Networks" in Proc. ICPPW 2005.

Mixing Heterogeneous Address Spaces in a Single Edge Network

Il Hwan Kim and Heon Young Yeom

School of Computer Science and Engineering,
Seoul National University,
Seoul, 151-742, South Korea
ilhwan@dcslab.snu.ac.kr, yeom@snu.ac.kr

Abstract. The growth of IPv4 Internet has been facing the infamous IP address depletion barrier. In practice, typical IPv4 Internet edge networks can be expanded by incorporating private addresses and NAT devices.

In this paper, major limitations of NAT-expanded private networks are presented. Furthermore, a solution is proposed to encourage the mixed usage of private and public IP addresses in a single edge network domain. The solution comprises of two key ideas : super-subnet mask and shared NAT. Super-subnet mask removes the routing boundary between private and public hosts. Shared NAT saves public IP address resources by sharing them among several private networks. These ideas not only encourage the coexistence of heterogeneous address classes, but also lead to efficient sharing of global IP addresses.

1 Introduction

1.1 NAT-Expanded Private Networks

As a prevailing and cost-effective data communication medium, Internet is now recognized as the infrastructure for communication. For instance, all-IP based home network, triple play, and nation-wide telematics are carefully explored as the most promising services despite of its not-totally-reliable nature. These application services need terminal devices to be connected to Internet; these devices need two to ten times more IP addresses than those actually in use today.

In practice, many IPv4 Internet edge networks are individually expanded by subscriber-initiated deployment of private addresses and NAT (Network Address Translator) [1] devices. In this paper, such a form of network will be called as NAT-expanded private network.

In a NAT-expanded private network, private hosts are normally invisible to and not accessible by public hosts outside the NAT. A traditional solution for free and unrestricted accesses to an internal host is the mixed usage of public IP addresses with the DMZ (De-Militarized Zone) [2] configuration.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 578–581, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1.2 Limitations of Legacy Mixed NAT-Expanded Networks

Mixing private and public hosts can incur cumbersome problems with regard to arrangement and inter-operability.

1. The NAT hinders private and public hosts from reaching to each other, by separating a premises network into two different routing domains.
2. The NAT consumes at least one public address per one premises network and the occupation tends to be long.
3. The NAT limits the performance of not only inter-premises but also intra-premises network. The public host must content with private hosts for the NAT(and routing) resources, in order to reach an intra-premises host.

Although the mixed NAT-expanded networks have these drawbacks, it can achieve wider deployment with much higher level of services, if a few fundamental improvements are applied.

2 Super-Subnet Mask and Shared NAT

2.1 Super-Subnet Mask

The most straightforward solution to the first problem – the lack of direct routes – is to assign an oversized subnet mask to local hosts. If a network address is large enough to cover all the private and public addresses in the premises, the subnet mask for the oversized network address is called a super-subnet mask[1].

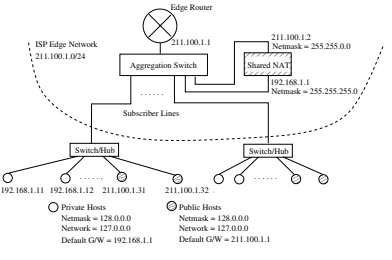


Fig. 1. A shared NAT in the super-subnet organization

An example of a super-subnet organization is illustrated in Fig. 1. In the super-subnet organization, the edge router is designated as the default router for the public hosts. And the NAT is assigned as the default gateway for the private hosts. No additional configurations are required in a super-subnet.

[1] To avoid misunderstandings with the word "supernet", a different name is adopted in this paper.

2.2 Shared NAT

When a private host wants to communicate with the global Internet hosts, its address should be translated into a public one by a NAT. Because the super-subnet organization removes the logical boundary between each premises network, a single NAT server is sufficient to serve all the private hosts within the edge network.

A shared NAT serves the private hosts with a route to the global Internet, in the similar way as an edge router serving the local public hosts. A large number of individual NATs can be replaced by only one shared NAT. Hence, the total number of public IP addresses occupied by NATs can be reduced significantly across an edge network.

2.3 Super-Subnet ARP Proxy

In Fig. 1, A and B consider 128.0.0.0/1 network is local, and 0.0.0.0/1 is outside. Because their view is erroneous and mismatches the real world, an ARP proxy must guide them to an appropriate router. The private host A must forward its outbound packets to a NAT, and the public host B must forward them to the edge router.

The special version of an ARP proxy is called a super-subnet ARP proxy. It is responsible to listen to all the ARP requests flowing through the edge network, and provides answers to them if necessary. The ARP proxies can be strategically placed to minimize the risk of ARP storm and the impact of misbehaving clients.

The ARP proxy classifies an ARP query according to Table 1, and then answers an appropriate reply to the originator. With an ARP proxy, any super-subnetted hosts can communicate with each other as well as with foreign hosts. Although the proof is simple, it is omitted due to the page limitation.

Table 1. Address resolution table of the super-subnet ARP proxy

	Destination		
Source	Local Public	Local Private	Foreign
Local Public	N/A	keep silent	edge router
Local Private	keep silent	N/A	shared NAT
Foreign	bypass	bypass	N/A

3 Conclusion

The advantage of super-subnet can be summed up as follows.

Compared to the legacy NAT-expanded networks, public IP addresses are less wasted and need not to be occupied all the time. The public IP resource can be saved for more precious servers instead of wasted by individual NATs.

All hosts within the super-subnet can directly communicate via L2 links. The inefficiency caused by individual NATs is ameliorate by direct routing between local hosts.

High performance bridge-based residential gateways can be implemented with lower costs compared to NAT-based ones. Contrary to that NAT-based RGs should be loaded with powerful processor and complicated softwares, bridge-based RGs can be implemented with a few simple extensions to the L2 switches. They are more competitive to the simple L2 switches than NAT-based ones, with respect to the performance.

4 Related Works

There are some related works with regards to address reuse and NAT extensions.

MobileNAT [3] utilizes the NAT techniques to facilitate the mobility of wireless IP networks. The extensive use of DHCP is also applicable to other NAT related works.

As a method for address sharing that exhibits more transparency than NAT, RSIP [4] is proposed and published as RFC 3103 [5]. RSIP requires hosts to be modified in order to interact with RSIP gateways.

Another point of view on the NAT problems is that it makes the peers confused about their identities by implicitly transforming the address headers. As a consequent, new routing and tunneling protocols that play the similar role as NAT are proposed. A few explicit identity-based routing mechanisms are known so far [6] [7] [8].

References

1. Srisuresh, P., Egevang, K.: Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational) (2001)
2. Srisuresh, P., Holdrege, M.: IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Proposed Standard) (1999)
3. Buddhikot, M., Hari, A., Singh, K., Miller, S.: Mobilenat: a new technique for mobility across heterogeneous address spaces. In: WMASH '03: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots, New York, NY, USA, ACM Press (2003) 75–84
4. Borella, M., Montenegro, G.: Rsip: Address sharing with end-to-end security (2000)
5. Borella, M., Grabelsky, D., Lo, J., Taniguchi, K.: Realm Specific IP: Protocol Specification. Internet Engineering Task Force: RFC 3103 (2001)
6. Ramakrishna, P.F.: Ipnl: A nat-extended internet architecture. In: SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2001) 69–80
7. Turányi, Z., Valkó, A., Campbell, A.T.: 4+4: an architecture for evolving the internet address space back toward transparency. SIGCOMM Comput. Commun. Rev. **33**(5) (2003) 43–54
8. Walfish, M., Stribling, J., Krohn, M., Balakrishnan, H., Morris, R., Shenker, S.: Middleboxes no longer considered harmful. MIT Technical Report TR/954 (2004)

Delivery and Storage Architecture for Sensed Information Using SNMP*

DeokJai Choi[1], Hongseok Jang[2], Kugsang Jeong[2],
Punghyeok Kim[2], and Soohyung Kim[1]

[1] Dept. Of Coputer Science, Chonnam National University
300 Yongbong-dong, Buk-gu, Gwangju, 500-757, Korea
{dchoi, shkim}@chonnam.ac.kr
[2] Dept. Of Coputer Science, Chonnam National University
300 Yongbong-dong, Buk-gu, Gwangju, 500-757, Korea
{jwang, handeum, dcwork}@iat.chonnam.ac.kr

Abstract. Many researches on context aware computing are carried out around the world. Among them, Context-Toolkit and Semantic Space provide separation of concerns between sensor and application. They make application developing easier. However, they have one problem that is lacking of simplicity in communication, compatibility and flexibility in building systems. To solve it, we propose one delivery and storage structure using standardized simple network management protocol which is useful to deliver, store and manage sensed information. We also verify that this architecture is efficient in wireless sensor network to deliver and store environmental information through an implementation of a SNMP agent. We confirm that this architecture with simplicity, compatibility and flexibility gives the efficiency to developing systems.

Keywords: sensed information management, sensor network management, SNMP.

1 Introduction

Many researches about context aware computing are carried out around the world. Among them, Context-Toolkit[1] and Semantic Space[2] use a medium to manage context information between the sensor and application in order to resolve the problem of dependency. However, to let the application use context information, the medium in between must be newly established, or a complex connection must be made with the previous medium.

In Context-Toolkit, there is a problem of increasing connection points for communications. An example for this can be the understanding of overall resources -

Toolkit must execute resource discovery to understand the overall resources. The widget of Context-Toolkit has a communication module and data-processing module, which are vulnerable to the application and communication protocol of

* This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 582 – 585, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Context-Toolkit. Since a fixed communication type must be followed in Context-Toolkit, flexibility in development was decreased.

In Semantic Space, a Context Wrapper, which is like the widget in Context-Toolkit, transmits sensed information to the application or aggregator. To add or delete sensors, Context Wrapper is added to or deleted from the context aware system called smart space through UPnP service. Here some problems of resource discovery and communication type compatibility also exist, similarly in Context-Toolkit.

Thus, by establishing a sensor platform using the transmitting and saving functions of the well-known network management protocol, SNMP[3], we have tried to improve the problems in Context-Toolkit and Semantic Space.

By securing the transparency of sensor through servicing wireless sensed information received by the host PC with a SNMP agent, and by managing the sensed information of wireless network, this paper shows the possibility of flexible transmission and saving of sensed information through SNMP.

2 Delivery and Storage Architecture

The composition of the architecture which proposes in this paper is same below Fig.1.

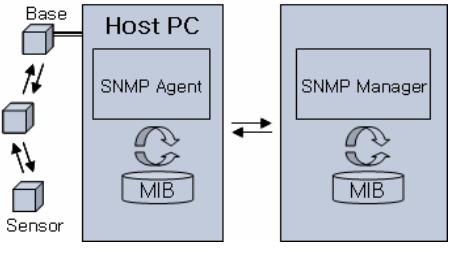


Fig. 1. Composition of architecture

SNMP agent of host PC, which collects and manages environment sensed information. And SNMP manager of outside systems, which can approach to information, approaches to SNMP agent.

SNMP agent module is consisted of SNMP agent, data transmittance, and MIB. SNMP agent manages and controls sensor. MIB stores environment information of sensor. SNMP manager module is consisted of SNMP manager and MIB.

In this paper, we compose server side's system (agent) to confirm that SNMP agent could serve environment information from wireless environments to client side's system (manager). Below Fig. 2 shows composition of agent part.

The sensor modules sense from environments and deliver sensed information to other sensor modules or the base one through wireless communications. Sensor modules install application programs managed by TinyOS. In this paper, we measure temperature, illumination, internal voltage etc. by installing a OscilloscopeRF

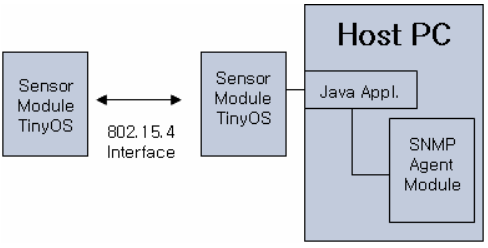


Fig. 2. Composition of Agent part

application program. Except the base sensor module, the others are installed in OscilloscopeRF program that could send the sensed values. The base sensor module is installed in base station program.

The base sensor module acting like a gateway passes the sensed information acquired by the UART using Java interface to the SNMP agent application module. We composed the system that passes the sensed information using general MoteIF application program here.

Environment information accepted in this host PC was passed to a SNMP agent part and the sensed values was managed by the SNMP agent. After monitored by an automatic polling program, we received the result managed by SNMP agent in this sensor information table, Fig.3. The periodic polling test shows that the SNMP agent brings environment information from the sensor properly.



Fig. 3. Monitoring of sensor table

The SNMP manager acts like a client collects environment information from the SNMP agent and manages sensors through these commands Trap, Get, Set etc.

SNMP is widely used all round as a standard protocol, and the several application programs are possible. SNMP manager can access to the SNMP agent through basic information (IP, community information, MIB OID etc.) that can approach to the MIB.

Simply inserting SNMP manager module to an application or a middleware will get the environmental information of the sensor easily and the information will be able to use. Fig.4 is the environment information of sensor got from MIB browser (client).

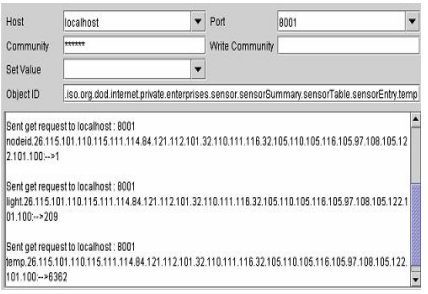


Fig. 4. Sensed information that uses Get command

Through the experiments above, I confirm that the SNMP module could search and control the environment information of sensor.

The wrapping of the agent using SNMP provides a transparent characteristic to the developers of application programs who do not need to know every vendor's API in input processing of various sensors. This gives the efficiency to developing systems.

3 Conclusion

In order to manage sensed information that supports wireless sensor network, this paper suggests a method of transmitting and saving sensed information through SNMP protocol. Also, through realizing agent parts, it suggests that the structure using previous SNMP protocol is also efficient in transmitting and saving sensed information. Through using such architectures, a sensor management structure can be built to satisfy the functions of information transmission, information saving, conditional informing and control.

By using a standardized SNMP protocol, problems in previous context aware computing technologies can be resolved, such as resource discovery, incompatibility and singleness of communication type, insufficient flexibility in development, etc.

References

1. Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications," PhD thesis, College of Computing, Georgia Institute of Technology, Dec 2000.
2. X. Wang, D. Zhang, J. S. Dong, C. Chin and S. R. Hettiarachchi. "Semantic Space: A Semantic Web Infrastructure for Smart Spaces", IEEE Pervasive Computing, 3(3):32-39, July-September 2004.
3. William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON1 and 2 Third Edition," Addison Wesley New York.

GISness System for Fast TSP Solving and Supporting Decision Making

Iwona Pozniak-Koszalka, Ireneusz Kulaga, and Leszek Koszalka

Chair of Systems and Computer Networks, Wroclaw University of Technology,
50-370 Wroclaw, Poland
leszek.koszalka@pwr.wroc.pl

Abstract. This paper shows that geographic information system can successfully solve TSP (travelling salesman problem). It has been done using a module of the designed and implemented by authors GISness system. Three algorithms for solving TSP are available, including the proposed by authors an hybrid algorithm called 2optGAM. The results of research show that the algorithm is very promising.

1 Introduction

Nowadays business organizations are gathering data about sales, customers and demographic profiles. Most of the collected data are geographic data with two important features: a value and its location in space. It was estimated in [5] that more than 70% of trade data is spatially oriented. Geographic information systems (GIS) allow collecting, manipulating and analyzing data in a spatial context.

The designed and implemented by authors GISness (GIS for Business) system is GIS system with elements of DSS (Decision Support System). A set of possible decisions is created as a result of user interaction with system. GISness offers three different views on geographic information: (i) the database view – every object on a digital map has a corresponding entry in DB, (ii) the map view – data are presented in form of interactive maps; maps allow querying, editing and analysing information, (iii) the model view – a sequence of functions used to transform data into information together with sets of data form a cartographic model [4, 5].

Since the TSP is spatially oriented, geographic information systems can serve as a sophisticated environment for TSP data manipulation and visualization [1, 10]. We created a module of GISness for solving TSP. Three approaches were adapted, including the well-known nearest neighbour algorithm, 2opt method [8], and our own algorithm 2optGAM being an improvement of the 2optGA algorithm [9].

2 GISness System

GISness has been developed in document-view architecture, where the document is a digital map containing information about objects. This map contains *spatial data* and

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 586 – 589, 2006.
© Springer-Verlag Berlin Heidelberg 2006

descriptive data. One of the main features of GISness is the possibility of linking database to the map. This distinguishes GISness from GIS-systems which use static maps and other graphic applications, e.g. CAD. GISness system stores descriptive data in database governed by its own efficient management system GISnessDB. Objects can be represented by attributes defined by the user as a quantity e.g. a number of inhabitants, or as a text e.g. a name of a town. GISness allows exchanging descriptive data with other database systems through import and export data.

The system was designed with UML standard supported by Power Designer 9.0 in Sybase environment [6]. It has been developing since 2004 as light but robust system [4]. Recently, it is composed of 21 000 lines of C++ code that create 82 classes.

GISness provides user with functions for *descriptive data analysis*: classification and/or selection of objects depending on their features, performance of calculations on attributes, viewing on attributes statistics and graphs, and for *spatial data analysis*: calculation of distances between objects, calculation of statistics depending on location and objects relations, and visualization of results of analysis.

3 Algorithms to Solving TSP

An example of using GISness to solving TSP is shown in Fig. 1. Three algorithms for solving TSP are available in GISness system.

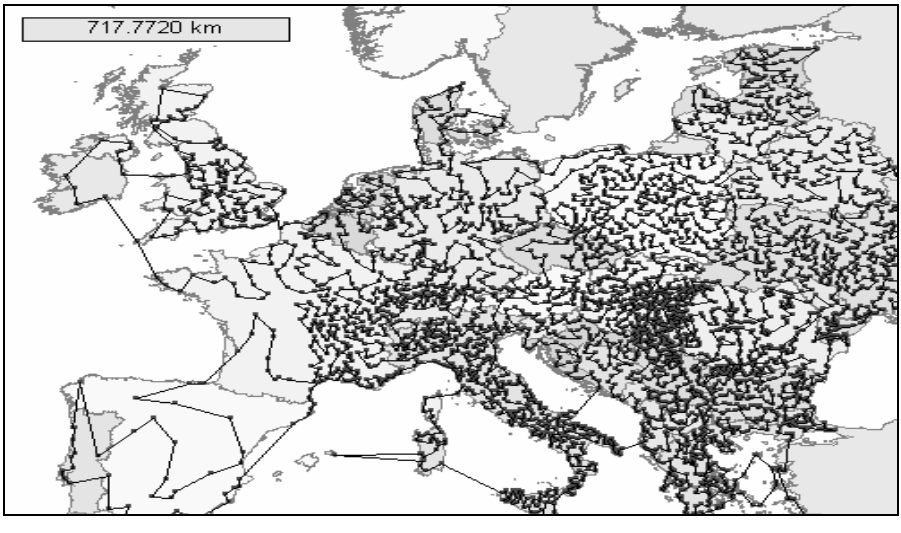


Fig. 1. A solution for 4152 cities problem using 2optGAM

The **nearest neighbour** [8] algorithm starts in a chosen city and in turn selects the closest unvisited city until all cities are in the route. Solutions often contain crossing edges and the length of route depends on a chosen starting point. GISness computes routes for each city taken as starting point and returns the shortest one as the solution.

The **2opt method** returns local minima in polynomial time [8]. It improves route by reconnecting and reversing order of subroutes. Every pair of edges is checked

whether an improvement is possible. The procedure is repeated until no further improvement can be done.

The proposed **2optGAM algorithm** is an hybrid algorithm. Combination of GA and heuristics are very common [2, 3, 9, 11] way of improving genetic algorithms. Although such a combination makes hybrid algorithm domain dependent, the performance is supposed to be increased [3]. The 2optGAM uses Greedy Subtour Crossover operator, however, the mutation operator is modified. If the chosen individual has been already improved by 2opt procedure, then two randomly selected genes are exchanged, and next, 2opt is being applied once again (Fig. 2).

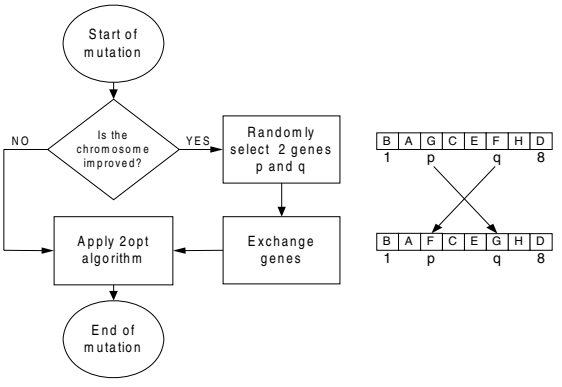


Fig. 2. Modified mutation operator

4 Investigations

A special experimentation system in Java was created (available on Website [12]). Apart from the opportunity of random generating cities, the system allows also loading common TSP problems from TSPLIB [7].

Experiment 1. In Table 1 the average results obtained by 2optGA and 2optGAM for a randomly generated problem (200 cities) in a specified time horizon are shown. The size of population was set to 200. Mutation and crossover probabilities were equal to suitable 0.3 and 0.2, respectively. The experiment was repeated 10 times.

Table 1. The results of algorithm comparison for randomly generated TSP of 200 cities

running time [s]	length of route						number of generations	
	2optGA			2optGAM			2optGA	2optGAM
	min	**avg**	max	min	**avg**	max	**Avg**	**avg**
30	3867	**3907**	3933	3808	**3841**	3869	**124**	**77**
60	3792	**3847**	3885	3792	**3803**	3814	**306**	**207**

Table 2. The results of comparison of 2optGA and 2optGAM algorithms for common TSP problems (the column *optimal** contains lengths of optimal routes from TSPLIB)

TSPLIB	optimal*	2optGA				2optGAM			
		avg	diff [%]	c	t [s]	avg	diff [%]	#G	t [s]
att48	33523	33523	0	129	0,8	33523	0	15	0,2
berlin52	7544	7544	0	77	0,6	7544	0	7	0,1
Kroa100	21285	21285	0	122	2,9	21285	0	33	1,4
Kroc100	20750	20750	0	167	3,8	20750	0	52	2,1
eil101	642	645	0,46	3942	68,7	642	0	103	3,6
lin105	14382	14382	0	93	2,6	14382	0	41	1,8
A280	2586	2624	1,46	1133	213,3	2603	0,65	712	179,4
Pr1002	259066	275723	6,42	824	2961,1	264675	2,16	1081	6001,9

The solutions returned by 2optGAM are better in both cases of running time.

Experiment 2. Table 2 presents solutions (the averaged lenghts of routes (avg) and parameters e.g. the total number of generations (#G), time duration t[s]) obtained for problems taken from TSPLIB. Algorithms were stopped after finding the optimal solution or if current best solution was not changed over a specified amount of time.

In most cases, both algorithms produced optimal solutions, however, it took less time for 2optGAM algorithm. Moreover, in more complex problems (a280, eil101) the 2optGAM returned not optimal solutions but better than 2optGA improvements.

It may justify the conclusion that the 2optGAM algorithm is very promising.

References

1. Abudiab, M., Starek M., Lumampao R. and Nguyen A.: Utilization of GIS and Graph Theory for Determination of Optimal Mailing Route. JCSC 4 (2004) 273-278
2. Bryant K., Benjamin A.,: Genetic Algorithms and the Travelling Salesman Problem. Working Paper, Departament of Mathematics, Harvey Mudd College (2000)
3. Cotta C., Adlana J.F., Troya J.M.: Hybridizing Genetic Algoriths with Branch and Bound Techniques. Artificial Neural Nets and Genetic Alg., Springer-Verlag (1995) 278-280
4. Kulaga I., Pozniak-Koszalka I.: Gisness System – Cartographic Modelling to Support Business Decision-Making. Proc. of 4th Polish-British Workshop, Ladek (2004) 93-102
5. Harmon J.E., Anderson S.J.: The Design and Implementation of Geographic Information Systems. John Wiley, New York (2003)
6. Pozniak-Koszalka I.,: Relational Data Bases in Sybase Environment. Modelling, Designing, Applications. WPWR, Wroclaw (2004)
7. Reinelt G.: TSPLIB – A Travelling Salesman Problem Library. ORSA Journal on Computing, 3 (1991) 376-384
8. Reinelt G.: The Travelling Salesman: Computational Solutions for TSP Applications. Springer-Verlag (1994)
9. Sengoku H.,Yoshihara I.: TSP Solver Using GA on Java. Proc. of AROB (1998) 283-288
10. Shaw S.: Transport Geography on the Web. http://people.hofstra.edu/geotrans/ (2005)
11. website: http://www.zlote.jabluszko.net/tsp/

A DNS Based New Route Optimization Scheme with Fast Neighbor Discovery in Mobile IPv6 Networks

Byungjoo Park and Haniph Latchman

Department of Electrical and Computer Engineering,
University of Florida, Gainesville, USA
{pbj0625, latchman}@ufl.edu

Abstract. In this paper, we propose a new route optimization scheme (EDNS-MIPv6) for MIPv6 using the enhanced Domain Name System (DNS) together with local mobility management scheme to reduce the signaling delay. EDNS-MIPv6 alleviates the triangle routing problem and reduces both the signaling delay to the home network, and the home agent's processing load.

1 Enhanced DNS Based Mobile IPv6 (EDNS-MIPv6)

1.1 Fast Neighbor Discovery and Local Mobility Management

We offer a fast neighbor discovery scheme to quickly process the DAD procedure in MIPv6. Our proposed scheme uses the look up algorithm in the modified neighbor cache of a new access router [5]. The DAD using lookup algorithm consumes an extremely short amount of time, typically a few micro second units, such as Longest Prefix Matching speeds in routing table. In Patricia Trie case, since lookup requires memory access of 48 times in worst case, the number of lookup value is 48. Hence, Lookup delay is 5.28 μ sec in worst case.

In the conventional DNS-based Mobile IP scheme the DNS update problem was resolved using the smooth handoff scheme. However, it resulted in data loss because it could not update quickly in an environment where the MN moves frequently. To solve this problem we can use local mobility management. When an MN moves to a foreign network the MN acquires a CoA through a router advertisement message from the new access router (AR). By constructing routers hierarchically using Mobility Anchor Point (MAP), we can minimize the delay produced when an MN updates its CoA in the DNS Resource Record.

As new system architecture could be cost prohibitive, we present a way to expand DNS. We use inverse domain which changes the name into the address and maps it. First, we separate a ufl.edu name server to act as a domain name caching system and an inverse domain system. For example, when an FTP server receives a packet addressed to 128.227.120.42 from an FTP client, the FTP server must check whether this FTP client is identified or not. The FTP server could reference the resource data files which have identified client lists; however, these

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 590–593, 2006.
© Springer-Verlag Berlin Heidelberg 2006

files include domain names only, so the FTP server has a "RESOLVER" which is the DNS client. The RESOLVER sends an inverse query asking the name of the FTP client to the DNS server and inverse domain system. Through this, the FTP server acquires the domain name of the FTP client. To support the domain name we modified the inverse domain to find out the MN's position in the DNS. The servers conducting an inverse domain have hierarchical levels. If an MN receives a new CoA, the MN sends the home address CoA and update request (registration) message including lifetime to the DNS. Then the DNS confirms the MN's home address and the DNS Resource Record. If a mobility biding for the MN did not exist, the DNS would either create a new resource record or update the existing one. If the CoA was found, only the lifetime would be updated. However, if the CoA was not found the existent CoA would be substituted and the lifetime would be updated as well.

1.2 Registration and Routing Procedure in EDNS-Mobile IPv6 Architecture

In this section we present the procedure of our proposed method. Step 1. When an MN moves to new foreign network, it establishes a path to the MAP using the path setup message. Then we can construct a DNS update message from the path setup message. Step 2. The DNS message includes the MAP's CoA, lifetime, and an MN's home address. The DNS's RR, which includes mobility bindings, is updated by the DNS update message. If an MN is in a foreign network before the lifetime elapses, the RR has to update continuously. Alternatively, when an MN moves to a new foreign network which uses a new MAP, the RR has to be updated with the new MAP's CoA. Movement within the same sub-network requires updating the lifetime only. Step 3. When the CN wishes to send packets to an MN, its resolver sends a query message to the DNS server. The DNS server checks the query message the reserved bits in order to identify the message's purpose as retrieving the MN's domain name or not. The reserved field uses three bits for the MN; therefore, when the reserved bits are "001," the MN does get the domain name. The Opcode is set to "0011" to specify that a query message is a mobile query. When the MN does not want the domain name, the reserved bits could be set to "002." With an Opcode of "0100". The Opcode field contains a 4 bit value that identifies the type of query the message contains. We use six different Opcode values such as '0: A standard normal Query', '1: An normal Inverse Query', '2: A Server Status Request', '3: A Mobile Query', '4: A Mobile Inverse Query' and '5-15: Reserved for future use'. Step 4. The DNS server has to make a new response message to manage the RR for an MN. The response record receives the 0xC00C pointer for the query record instead of the domain name. Then, the DNS sends the MN's CoA, lifetime, and routing information to the CN. Step 5. The CN can send a packet addressed to an MN directly, without tunneling, via the HA. When packets sent by the CN arrive in the MAP, the MAP sends the packets to the MN after checking the MAP's visitor list and using the MN's home address found in the packet header.

2 Analysis of the Delay Needed to Transmit Packets from the CN to MN Directly

In this section we analyze the delay required for the DNS name resolution. We separate he signaling procedure for packet delivery in two cases: registration procedure and routing procedure;

2.1 Standard MIPv6 Signaling Procedure

(a) Registration Procedure (1∼7) 1: MN moves from a Previous Foreign network to a New Foreign Network: t_{L2}. 2: MN sends a Router Solicitation message (RS) to the NAR: t_{RS}. 3: MN receives the prefix information about the NAR included in the RA message: t_{RA}. 4: MN makes a new CoA configuration using stateless auto configuration: t_{CoA}. 5: NAR begins processing DAD to determine the uniqueness of the newly generated CoA: t_{DAD}. 6: MN registers the new CoA to the HA: t_{BU-HA}. 7: HA caches the MN's new CoA in the HA's buffer: t_Q

(b) Routing Procedure (8∼11) 8: CN sends a packet to the MN's home address: t_{CN-HA}. 9: HA starts intercepting all packets addressed to the MN and the packets are tunneled from the HA to the MN's new CoA: $t_{HA-NCoA}$. 10: MN sends a binding update message with the MN's new CoA to the CN after receiving the first tunneled packet from the HA: t_{BU-CN}. 11: CN sends packets directly to the MN using the new CoA: t_{Packet}

2.2 Proposed EDNS MIPv6 Signaling Procedure

(a) Registration Procedure (1∼7) 1: MN moves from a Previous Foreign network to a New Foreign Network: t_{L2}. 2: MN makes a new tentative CoA configuration using stateless auto configuration with an SRA message: t_{T-CoA}. 3: MN sends a Router Solicitation message (NRS) to the NAR: t_{NRS}. 4: NAR begins processing DAD to determine the uniqueness of the newly generated tentative CoA using the LookUp algorithm [3]: t_{LU}. 5: MN receives the prefix information about the NAR included in the RA message: t_{NRA}. 6: MN registers a new CoA to the HA: t_{BU-HA}. 7: HA caches the MN's new CoA in the HA's buffer: t_Q

(b) Routing Procedure (8∼10) 8: HA send a DNS update message to the Fixed DNS located in the home network to update the MN's CoA: t_{DNS-UP}. 9: CN interrogates the Fixed DNS in order to know the IP address of the MN with which it wishes to communicate: t_{DNS}. 10: CN sends packets directly to the MN using the new CoA: t_{Packet}.

The total registration delay for standard Mobile IPv6 and proposed EDNS-MIPv6 are presented as $T_{Reg-MIPv6} = t_{L2}+t_{RS}+t_{RA}+t_{CoA}+t_{DAD}+t_{BU-HA}+t_Q$ and $T_{Reg-EDNS} = t_{L2}+t_{T-CoA}+t_{NS}+t_{LU}+t_{NRA}+t_{BU-HA}+t_Q$ respectively.

The total routing delay for standard Mobile IPv6 and proposed EDNS-MIPv6 are presented as $T_{Routing-MIPv6} = t_{CN-HA} + t_{HA-NCoA} + t_{BU-CN} + t_{Packet}$ and $T_{Routing-EDNS} = t_{DNS-UP} + t_{DNS} + t_{Packet}$ respectively.

Therefore, the total signaling delay ($T_{Signaling}$) for packet delivery between the CN and an MN is the sum of the registration delay (T_{Reg}) and the routing delay ($T_{Routing}$). First of all, we assume that the mobile node receives the IP address without a domain name. With "$T_{Signaling-MIPv6} = T_{Reg-MIPv6} + T_{Routing-MIPv6}$" and "$T_{Signaling-EDNS} = T_{Reg-EDNS} + T_{Routing-EDNS}$", we can find the delay difference, T_{Diff}, between these two mechanisms.

$$T_{Diff} = \{T_{Signaling-MIPv6} - T_{Signaling-EDNS}\} = (t_{DAD} - t_{LU}) + t_{HA-NCoA} + (t_{CN-HA} + t_{BU-CN} - t_{DNS-UP} - t_{DNS}) \quad (1)$$

From Eq.1, T_{Diff} is always a nonnegative integer because "$t_{CN-HA} + t_{BU-CN}$" is greater than or equal to "$t_{DNS-UP} + t_{DNS}$" in wireline-network environment. Also, since t_{DAD} is larger than t_{LU}, when an MN moves to a new network area the value of T_{Diff} is always positive. What is inferred from "$T_{Routing-MIPv6}$" is that the important factor affecting the delay is the packet delivery time during tunneling from the HA and the MN's new CoA, $t_{HA-NCoA}$. If this tunneled packet delivery time is long, the total delay increases. Therefore, packets sent from the CN are delivered via the HA and tunneled to the MN until the CN should receives a binding update message from the MN. Thus, by using the EDNS-MIPv6 scheme we reduce the total signaling delay and remove the triangle routing problem which causes degradation of network performance.

3 Conclusion

In this paper, we have proposed a new route optimization scheme (EDNS-MIPv6) for MIPv6 using the enhanced Domain Name System (DNS) together with fast neighbor discovery and local mobility management scheme to reduce the signaling delay using lookup algorithm. EDNS-MIPv6 alleviates the triangle routing problem and reduces both the signaling delay to the home network, and the home agent's processing load.

References

1. C. Perkins, K-Y. Wang. "Optimized smooth handoffs in Mobile IP", Proceedings of IEEE Symposium on Computers and Communications, Egypt, July 1999.
2. M.Conti, E.Gregori, S.Martelli, "DNS-based architectures for an efficient management of Mobile Users in Internet ", IEEE, 2001
3. Byungjoo Park, Sunguk Lee, Haniph Latchman, "Performance Analysis of Enhanced Mobile IPv6 with Fast Handover over End to End TCP", in. Proc. of IEEE Wireless Com-munications and Networking Conference (WCNC), April 3-6, 2006.

Performance Analysis of Group Handoff in Multihop Mesh Relay System

Young-uk Chung, Yong-Hoon Choi, and Hyukjoon Lee

Kwangwoon University, Seoul, Republic of Korea
yuchung@kw.ac.kr

Abstract. Simulcast Technique can be used to diminish handoff rate and share resources among multiple relays in MMR system. Using dynamic grouping, simulcast technique can also rearrange traffic load. In this paper, we analyze group handoff generated by dynamic grouping in MMR system. Performance is evaluated in view of blocking and handoff call dropping probability.

1 Introduction

The multihop mesh relay(MMR) is a promising solution to expand coverage, enhance throughput and system capacity to wireless broadband multimedia services such as IEEE 802.16 systems. The gains in coverage and throughput can be leveraged to reduce total deployment cost for a given system performance requirement and thereby improve the economic viability of IEEE 802.16 systems. In this system, all resources such as Channel Elements(CE) and call processing hardwares are located in the CS. In the CS, call processing hardwares and resources are shared among multiple relays and this can improve a trunking efficiency. However this makes the increase of handoff in the system. Also there are many hot-spot cells because of the variety of different moving patterns of subscribers. To solve these problems, the simulcast technique can be adopted [1]. In case that low traffic load arises, multiple relays can be grouped to share resources. All relays within a group broadcast and receive signals as if they are in the same cell. This grouping of relays can be dynamically changed according to a traffic distribution. By rearranging the simulcast group dynamically, the system can reduce handoff rates and protect the outbreak of hot-spot cell. When groups are reorganized by dynamic group simulcasting, a relay which is located in a group can be transferred to the other group. In this case, all calls in that relay must be handoffed simultaneously to the target group. This type of handoff is named 'Group Handoff'. In this paper, we analyze the performance of group handoff in MMR system.

2 System Modeling and Analysis

In this system, a CS manages all Channels of relays which are controlled by the CS. The analysis of group handoff is mainly affected by dynamic grouping

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 594–597, 2006.
© Springer-Verlag Berlin Heidelberg 2006

algorithm. In this paper, we analyze using the simplest dynamic grouping algorithm which sets up new grouping due to the number of available channels in each group. We define that the system consists of two simulcast groups which are controlled by a CS. Each group has limited number of channel, C, which is the maximum number of simultaneous calls that can be serviced in the group. We define that initial number of relays in group 1 and group 2 are N_1 and N_2, respectively. We define that there should be at least two relays in a group after group handoff. Also, we assume that there is no handoff arrival from the neighboring system. Dynamic grouping is executed because of nonuniform traffic distribution of each group due to different call arrival in each group. We assume that each group has different new call arrival rate and it is changed after new grouping is established. We suppose that relays in a group have the same new call arrival rate. Let new call arrival rate in group 1 and group 2 be λ_{n1} and λ_{n2}, respectively. Let handoff call arrival rate from neighbor groups be λ_h and group handoff generation rate be λ_g. We assume that call duration time, T_c is exponentially distributed with mean μ_c^{-1}. Let cell dwell time be T_d which is exponentially distributed with mean μ_d^{-1}. We define two threshold values to decide timing of grouping and select relay adequate to be handoffed: TH_h is upper threshold and TH_l is lower threshold. We divide total capacity of a group into three levels. If state of a group is Level 1, only group handoff-out can be generated. If a group is in state of Level 2, this group can receive handoffed relay from other group and send a relay to other group. In state of Level 3, only group handoff-in can occur.

Using birth-death process, we can derive the state transition diagram in view of total system[2]-[3]. Each state is assigned to s and the state of this process is defined as $s = (i, j, k, l)$ where i and k are the number of communicating calls in group 1 and group 2, respectively. Also, j and l are the number of relays in group 1 and group 2, respectively. There are five sets of state-transition. They are symbolized by character from A to E. X and $\bar{\mathsf{X}}$ make a reversible pair. We define the transition to right direction be X1 and the transition to left direction be X2 if the set of transition is X. A1 and B1 mean that a new call is originated in group 1 and group 2, respectively. When a relay is handoffed to other group, call arrival rate of the relay is not changed. So, the total call arrival rate of group 1, λ_1 is calculated as $\lambda_{n1} + (j - N_1)\frac{\lambda_{n2}}{N_2}$ when $(j \geq N_1)$, and $\frac{j}{N_1}\lambda_{n1}$ when $(j < N_1)$. Also, the total call arrival rate of group 2, λ_2 is given by $\lambda_{n2} + (l - N_2)\frac{\lambda_{n1}}{N_1}$ when $(l \geq N_2)$ and $\frac{l}{N_2}\lambda_{n2}$ when $(l < N_2)$. A2 and B2 mean that a call in group 1 and group 2 are ended, respectively.

D1 is the transition when a relay in group 1 is handoffed out to group 2. The number of calls in handoffed relay, n is defined as

$$n = \{ \lfloor \frac{i}{j} \rfloor - x,\ \lfloor \frac{i}{j} \rfloor - (x-1),\ \cdots,\ \lfloor \frac{i}{j} \rfloor,\ \cdots,\ \lfloor \frac{i}{j} \rfloor + (x-1),\ \lfloor \frac{i}{j} \rfloor + x \}. \quad (1)$$

There are $(2x + 1)$ cases of group handoff, where x is the variation factor. If we assume that occurring probability of each value is the same, the transition rate of group handoff is expressed as $\frac{1}{j} \cdot \frac{1}{2x+1} \cdot \lambda_g$. The first term, $1/j$ means the

probability of select one relay in a group. The term, $1/(2x+1)$ is the probability that the selected relay has value of n. We assume that λ_g has three sorts of values, λ_{g1}, λ_{g2}, and λ_{g3} according to events. λ_{g1} has the largest value. If "Level 1 $\Rightarrow$ Level 3" event occurs, λ_{g1} is adopted. When "Level 1 $\Rightarrow$ Level 2" and "Level 2 $\Rightarrow$ Level 2" events occur, λ_{g2} is adopted. In the event of "Level 2 $\Rightarrow$ Level 3", λ_{g3} is adopted.

E1 means the case that a group handoff is occurred from group 2 to group 1. From the same way of the case of D1, Each n and transition rate is expressed as

$$n = \{ \lfloor \frac{k}{l} \rfloor - x,\ \lfloor \frac{k}{l} \rfloor - (x-1),\ \cdots,\ \lfloor \frac{k}{l} \rfloor,\ \cdots,\ \lfloor \frac{k}{l} \rfloor + (x-1),\ \lfloor \frac{k}{l} \rfloor + x \} \quad (2)$$

$$\text{transition rate} = \frac{1}{l} \cdot \frac{1}{2x+1} \cdot \lambda_g. \quad (3)$$

We evaluate the performance of proposed scheme in view of blocking and handoff call dropping probability. Let $P(s)$ be the steady-state probability of state, s. A new call is blocked when all channels in the group are occupied. This case occurs when $i \geq C$ or $k \geq C$. And the blocking probability is given by

$$P_B = \sum_{j=1}^{N_1+N_2-1} \sum_{l=1}^{N_1+N_2-1} \sum_{i=0}^{C} P(i,j,C,l)|_{j+l=N_1+N_2} \quad (4)$$
$$+ \sum_{j=1}^{N_1+N_2-1} \sum_{l=1}^{N_1+N_2-1} \sum_{k=0}^{C} P(C,j,k,l)|_{j+l=N_1+N_2} - P(C,j,C,l)|_{j+l=N_1+N_2}\ .$$

As we assume that there is no queue and reservation channel for handoff call, blocking probability and handoff call dropping probability have the same value.

3 Numerical Results

We assume that x has the value of 1. And the number of calls in the relay, n have three types of value. We investigate several numerical examples in case of $\mu_c = 0.01$, $\mu_d = 0.03$, $C = 20$, and $N_1 = N_2 = 4$. To evaluate the performance of group handoff in this system, We consider two conditions. First, we generate numerical results with changing the disparity of call arrival rate in each group. We assume that $\lambda_{g1} = 0.9$, $\lambda_{g2} = 0.5$, $\lambda_{g3} = 0.1$, $TH_h = 16$, and $TH_l = 4$. We compare performance in case of executing dynamic grouping and fixed grouping. The result is shown in Fig. 1. In Fig. 1, D and F means the case using dynamic grouping and fixed grouping, respectively. $a : b$ means the fraction of new call arrival rate in group 1 and group 2. From the result, blocking probability and handoff call dropping probability of dynamic grouping has similar value in case that the difference of call arrival rate in each group is changed. But, in case of fixed grouping, we could see that the larger the difference of call arrival rate in each group, the higher the blocking probability and the handoff call dropping probability.

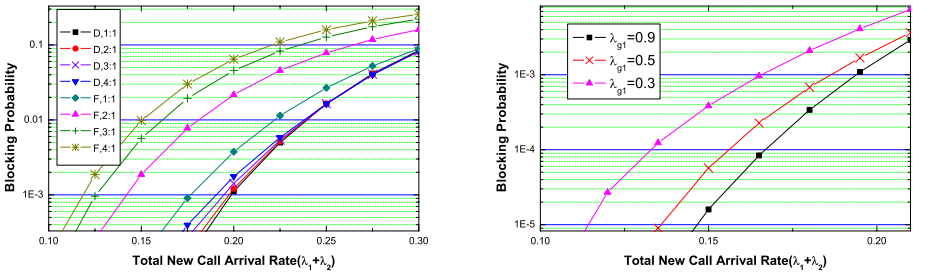


Fig. 1. Performance of Group Handoff

The second condition is the case that the values of λ_g are changed. In this case, we assume that $TH_h = 16$, $TH_l = 4$, and the fraction of new call arrival rate in group 1 and group 2 is 2 : 1. The result is also shown in Fig. 1. In Fig. 1, $\lambda_{g1} = 0.9$ means that $(\lambda_{g1}, \lambda_{g2}, \lambda_{g3}) = (0.9, 0.5, 0.1)$. $\lambda_{g1} = 0.5$ means that $(\lambda_{g1}, \lambda_{g2}, \lambda_{g3}) = (0.5, 0.3, 0.1)$. And, $\lambda_{g1} = 0.3$ means that $(\lambda_{g1}, \lambda_{g2}, \lambda_{g3}) = (0.3, 0.2, 0.1)$. The results show that the higher the value of λ_g, the lower the blocking probability and the handoff call dropping probability. That's because high value of λ_g can generate group handoff when it is needed. So, the blocking probability is diminished when the value of λ_g is high.

4 Conclusions

The MMR system was proposed to to expand coverage, enhance throughput and system capacity. In this system, simulcast technique using dynamic grouping can rearrange traffic load easily. Group handoff is generated by dynamic grouping. In this paper, we modeled group handoff using the Markov chain and performed system analysis. From numerical results, we can see that the larger the difference of call arrival rate in each group, the better the performance of dynamic grouping. Also, numerical result shows that dynamic grouping provides better performance than fixed grouping in simulcast technique.

References

1. S. Ariyavisitakul, T. E. Darcice, L. J. Greenstein, M. R. Philips and N. K. Shankaranarayanan, "Performance of Simucast Wireless Techniques for Personal Communication Systems," *IEEE JSAC*, Vol. 1, No. 4, pp.632-643, May 1996.
2. S. S. Rappaport, "The Multiple-Call Hand-Off Problem in High-Capacity Cellular Communication Systems," *IEEE Trans. on Vehicular Technology*, Vol. 40, No. 3, pp.546-557, Aug. 1991
3. S. -L. Su, J. -Y. Chen and J. -H. Huang, "Performance Analysis of Soft Handoff in CDMA Cellular Networks," *IEEE JSAC*, Vol. 14, No. 9, pp.1762-1769, Dec. 1996.

DSMRouter: A DiffServ-Based Multicast Router*

Yong Jiang

Graduate School at Shenzhen, Tsinghua University,
Shenzhen, Guangdong 518055, P.R. China
jiangy@sz.tsinghua.edu.cn

Abstract. In this paper, we realize a DiffServ-based Multicast Router (DSMRouter) to provide the QoS adjustment for multicasting video transmission in the DiffServ network. To reach the QoS goals, the proportional fairness scheduling strategy is proposed for layered media flows, including I-frame, P-frame and B-frame flows, to determine the approximate reserved bandwidth. Additionally, the DSMRouter system dynamically changes the sending rates of various service queues to ensure that layered media packets with higher priority are always sent before those with lower priority. The results reveal that DSMRouter dynamically modify layered media's sending rates among different service classes according to the network situation. The experimental results of the DSMRouter system on DiffServ networks also reveal that layered media with higher priority have less packet loss.

1 Introduction

The Internet Engineering Task Force (IETF) has proposed the Differentiated Service (DiffServ) architecture [1][2] to solve the Quality of Service (QoS) for different users and numerous types of traffic.

As we know, in an MPEG-coded video system, packet loss causes unexpected degradation of the quality of the received video, 3% packet loss in an MPEG-coded bit stream can be translated into a 30% frame error rate [3]. Shin et al. proposed a content-based packet video forwarding mechanism on the DiffServ network [4][5]. In [6], Striegel et al. proposed an architecture for DiffServ-based multicast that achieves the benefits of traditional IP multicast but does not require per group state information in the core routers.

In this paper, we realized a DiffServ-based Multicast Router (DSMRouter) based QoS model, for multicasting video transmission in the DiffServ network. The layered method [7] is used to transmit video streams, including I/P/B frames. One video stream is separated into several types of media flows using so-called layered media. When layered media are transmitted with different

* This research was sponsored by NSFC (No. 60503053) and GDNSF (No. 034308), and Development Plan of the State Key Fundamental Research (973) (No. 2003CB314805).

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 598–601, 2006.
© Springer-Verlag Berlin Heidelberg 2006

DSCP values, DSMRouter dynamically modifies layered media's sending rates among different service classes according to the network situation. DSMRouter guarantees that layered media with higher priority have less packet loss and so exhibit improved presentation to end users when the traffic is heavy.

2 Proportional Fairness Scheduling Strategy for Layered Media

Internet users and applications have diverse service expectations to the networks, making the current *same-service-to-all* model inadequate and limited. In the *relative differentiated services* [8] approach, the network traffic is grouped in a small number of *service classes which are ordered based on their packet forwarding quality*, in terms of per-hop metrics for the queueing delay and packet loss. In [9], we proposed the *proportional fairness principle*. According to this principle, the basic performance measures for packet forwarding locally at each hop are rated proportionally to certain *class differentiation parameters* that the network operator chooses, *independent of the class loads*.

In [10], we proposed the complete scheduling policy is described as follows:

Theorem 1. (Loss Ratio Proportional Fairness)
Consider a router that guarantees the delay proportion function $P_i^D(\cdot)$ for the class i and suppose that the input traffic R_i^{in} is b-smooth. If the buffer space allocated to the class is B_i and

$$B_i = \max\{b_i(t)(1-\sigma_i\tilde{l}) - P_i^D(t), t \geq 0\} \tag{1}$$

then the router guarantees the loss ratio proportional fairness principle.

Since the summation of buffer space B_i for all class can not be more than the total buffer space B_{total}, i.e. $\sum_{i=1}^{M} B_i \leq B_{total}$, the loss ratio proportional fairness parameter $\tilde{l}$ should satisfy $\tilde{l} \geq \frac{\sum_{i=1}^{M}[b_i(t)-P_i^D(t)]-B_{total}}{\sum_{i=1}^{M}\sigma_i b_i(t)}$.

Definition 1. (PFS strategy)
In term of equation, each service class is allocated appropriate buffers, and a arrival packet is dropped or enqueued according to available buffer space and packet dropper. Each waiting packet is assigned a deadline and packets are served earliest deadline first in a work-conserving manner. Specifically, in a given slot u there are $\sum_{i=1}^{M}(Q_i[u-1] + R_i^{in}[u])$ packets available for departure, and up to c packets with the smallest deadlines are selected for departure in slot u. The deadline D_i assigned to a packet from class i, which arrives in slot u is given by

$$D_i = \min\{t : t \geq u \quad and \quad Z_i(t; \quad u-1) \geq n_i\} \tag{2}$$

where

$$Z_i(t;u) = \min_{s:\tau(u)\leq s\leq u, Q[s]=0}\{R_i^{out}[\tau(u)+1, s] + P_i^D(t-s)\} \tag{3}$$

and n_i is the arrival count of the packets.

3 System Architecture

A DiffServ-based Multicast Router (DSMRouter) can be deployed at the edge of the DiffServ domain. DSMRouter can classify, policy and shape the traffic that transmit through it, just as one general DiffServ edge node. DSMRouter can provide different QoS guarantees according to the Per Hop Behavior (PHB).

A traffic sending mechanism under the PFS strategy, for streaming video, operates to enable transmitted streams between different DiffServ service classes to guarantee different QoS levels of the DiffServ network. The video stream is split into four classes-I frame, P frame, B frame and audio streams. Different classes have different delay bounds, loss bounds and bandwidth requirements. The delay, the packet loss and the bandwidth requirements of different frame streams can be analyzed to mark their DSCPs (Differentiated Service code-point) with different QoSs. In the DiffServ network, the layered video streams are sent by multicast. The frame streams are sent in the same multicast session with different port numbers and DSCP values.

4 Performance Evaluation

In this section, we have implemented our traffic sending mechanism under the PFS strategy in the DiffServ network, and evaluated the performance of the layered media scheme that is used in the DSMRouter.

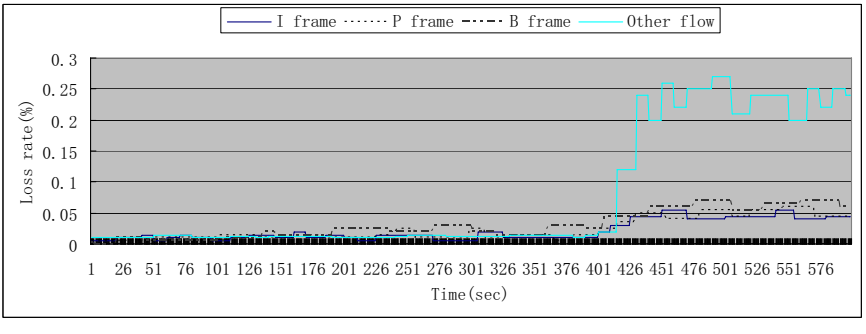


Fig. 1. Media flows-loss rates

Fig. 1 shows the loss rate distribution of I/P/B layered media streams. Clearly, PFS strategy achieves the loss rate proportional differentiation well. In 0 to 400 sec, media flows are transmitted through the DSMRouter with the PFS traffic sending mechanism. Under light load situation, the loss rate of I/P/B indicates that almost no data are lost with such a bandwidth reservation. While after 400 sec, the addition of the heavy load flow causes the loss rate of I/P/B to start to burst, but the layered media with higher priority have less packet loss and so exhibit improved presentation to end users when the traffic is heavy.

5 Conclusion

This paper has proposed a DiffServ-based media gateway called the DiffServ-based Multicast Router (DSMRouter) to guarantee QoS and scalability for multimedia presentation. In the reserving bandwidth situation, the approximate amounts of reserved bandwidth are computed for different service queues by applying the PFS strategy. With regard to the dynamic bandwidth adjustment situation, DSMRouter can be con-figured to serve different classes of layered media with various resource reservations based on the PFS strategy. The results of the performance analysis indicate that the system can detect the congestion of the network and react by immediately implement-ing appropriate procedures.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z.Wang, andW.Weiss, "An Architecture for Differentiated Services," RFC 2475, IETF, Dec. 1998.
2. P. Trimintzios, I. Andrikopoulos, G. Pavlou, C. F. Cavalcanti, D. Goderis, Y. T'Joens, P. Georgatsos, L. Georgiadis, D. Griffin, C. Jacquenet, R. Egan, and G. Memenios, "An archi-tectural framework for providing QoS in IP differentiated services networks," in Proc. 7th IFIP/IEEE Int. Symp. Integrated Network Management (IM 2001), May 2001, pp. 17-34.
3. J. Boyce and R. Gaglianello, "Packet loss effects on MPEG video sent over the public internet," in Proc. 6th ACM Int. Conf. Multimedia, Sep. 1998, pp. 181-190.
4. J. Shin, J. W. Kim, and C.-C. J. Kuo, "Content-based packet video forwarding mechanism in differentiated service networks," in Proc. Int. Packet Video Workshop, May 2000.
5. J. Shin, J. W. Kim, and C.-C. J. Kuo, "Quality of service mapping mechanism for packet video in differentiated services network," IEEE Trans. Multimedia, vol. 3, no. 2, pp. 219-231, Jun. 2001.
6. A.Striegel, G. Manimaran, "A Scalable Approach for DiffServ Multicasting", Proc. of Inter-national Conference on Communications, Helsinki, Finland, June 2001.
7. M. Kawada, K. Nakauchi, H. Morikawa, and T. Aoyama, "Multiple streams controller for layered multicast," in Proc. IEEE Int. Conf. Communications, vol. 1, 1999, pp. 65-68.
8. Dovrolis C, Stiliadis D. Relative Differentiated Services in the Internet: Issues and Mecha-nisms. In: Proceedings of ACM SIGMETRICS'99, Atlata , May 1999. pp204-205
9. Y. Jiang, C. Lin, and J. Wu. Integrated performance evaluating criteria for network traffic control. In: Proceedings of IEEE symposium on Computers and Communications 2001, IEEE Communications Society Press, Tunisia, July 2001.
10. Y. Jiang, J. Wu, The proportional fairness scheduling algorithm on multi-classes, Science in China Series F, Vol.46 No.3, June 2003, p 161-174.

Author Index

Abdalla Jr, H. 441
Abdurakhmanov, Abdurakhmon 221
Ahmed, Bilal 513
Ahn, Sang Ho 505
Aida, Masaki 461
Akbar, Ali Hammad 170
Alam, Muhammad Mahbub 1
Amad, Mourad 342
Amin, Syed Obaid 263
Ata, Shingo 322

Becker Westphall, Carlos 542
Beppu, Yasuyuki 33

Cha, Si-Ho 43
Chang, Byung-Soo 393
Chang, Yong 162
Chaudhry, Shafique Ahmad 170
Chen, Jui-Chi 412
Chen, Wen-Shyen E. 412
Cho, Kuk-Hyun 43
Choi, Bong Dae 162
Choi, Deok-Jae 383
Choi, DeokJai 582
Choi, Jaehyun 570
Choi, Jun Kyun 112, 403
Choi, Seong Gon 112, 403
Choi, Taesang 63
Choi, Yanghee 554
Choi, Yong-Hoon 594
Choi, YoungJun 574
Chong, Kiwon 570
Choo, Hyunseung 481, 525
Chu, Yul 517
Chugo, Akira 451
Chung, Hee Chang 422
Chung, Min Young 525
Chung, Young-uk 594
Crispim, H.A.F. 441

Dai, Tran Thanh 546

Gava Menezes, Alexandre 542
Gurin, Ivan 509

Ha, Ilkhyu 574
Ha, SangYong 332
Han, Chang-Hee 200
Han, Jeongsoo 190
Han, SunYoung 332
Han, Youngjoo 200
Hasegawa, Takaaki 293
Hirokawa, Yutaka 53
Hong, Choong Seon 1, 263, 513, 546
Hong, Chung-Pyo 102, 132
Hong, Daniel W. 393
Hong, James Won-Ki 242
Horiuchi, Hiroki 352
Hwang, InSeok 362
Hwang, You-Sun 180

Imai, Satoshi 451
Inoue, Masateru 210
Inoue, Takashi 210
Ishibashi, Keisuke 53

Jang, Hongseok 582
Jang, Won-Joo 132
Jeon, Byung Chun 112
Jeong, Chulho 372
Jeong, Jongpil 525
Jeong, Kugsang 582
Jeong, San-Jin 200
Jiang, Yong 598
Jin, Donxue 574
Ju, Hong Taek 242
Jung, Byeong-Hwa 491
Jung, Jae-il 558
Jung, Jaeyoun 574
Jung, Sunwoo 570

Kang, Dongwon 63
Kang, Seok-Joong 43
Kang, Tae Gyu 534
Kang, Tae-Hoon 102, 132
Kim, Chang-Hwa 491
Kim, ChinChol 332
Kim, Chonggun 574
Kim, Chulsoo 505

Kim, Dae Ho 534
Kim, Do Young 534
Kim, Dong-Hui 562
Kim, Dong Il 422
Kim, Dongkyu 570
Kim, Hyung-Soo 312
Kim, Il Hwan 578
Kim, Il-Yong 253
Kim, Jae-Hyun 180
Kim, Jae-Myung 283
Kim, Ki-Chang 253
Kim, Ki-Hyung 170
Kim, Ki Young 263
Kim, Meejoung 11
Kim, Punghyeok 582
Kim, Sangwan 63
Kim, Seong-Il 393
Kim, Shin-Dug 102, 132
Kim, Soohyung 582
Kim, Sungchun 521
Kim, Sungwook 521
Kim, Tae Ok 162
Kim, Taewan 505
Kim, Yoon Kee 471
Kim, Yoo-Sung 253
Kim, YoungJae 332
Kim, Young-Tak 162, 221, 312, 509
Kimura, Shingo 53
Kobayashi, Atsushi 53
Koo, Han-Seung 283
Koo, Jahwan 481
Koo, Ki Jong 534
Koszalka, Leszek 586
Kulaga, Ireneusz 586
Kwak, Kyungsup 122, 142
Kwon, Hye-Yeon 180
Kwon, Taeck-geun 82

Latchman, Haniph 590
Lee, Eunseok 372, 566
Lee, Hoon 471
Lee, Hyukjoon 594
Lee, Il-Kyoo 283
Lee, Jae-Oh 43, 562
Lee, Jong-Eon 43
Lee, Jong Hyup 422
Lee, Jong Min 112
Lee, Joonkyung 63
Lee, Kwang-Hui 471
Lee, Kyeongho 63
Lee, Soong Hee 422
Lee, Youngseok 82
Lim, Hoen-In 550
Liu, Yu 23

Mamun-or-Rashid, Md. 1
Mantar, Haci A. 152
Maruta, Toru 352
Matsubara, Daisuke 53
Matsuura, Hiroshi 302
Meddahi, Ahmed 342
Miyoshi, Yu 431
Mukherjee, Atanu 538
Murtaza, Syed Shariyar 513

Nagai, Koichi 322
Nah, Jae-Hoon 200
Nakamichi, Koji 451
Nascimento, Anderson C.A. 441

Oh, Haeng Suk 422
Oh, Sung-Min 180
Oka, Ikuo 322
Otsuka, Yoshihiro 431

Park, Ae-Soon 180
Park, Byungjoo 590
Park, Choon-Gul 550
Park, Jeongmin 372, 566
Park, Ju-Hee 550
Park, Seungchul 554
Park, Soo-Hyun 491
Pastor, Eduardo T.L. 441
Pathan, Al-Sakib Khan 546
Phu, Phung Huu 232
Pozniak-Koszalka, Iwona 586

Quyen, Le The 501

Ra, Sung-Woong 283
Rhee, Kyung Hyune 273
Ryu, Won 403

Saitou, Motoyuki 53
Sakamoto, Hitoaki 53
Sato, Masataka 210
Seo, Jong-Cheol 312

Seok, Seung-Hak 550
Seok, SeungHak 362
Shakhov, Vladimir V. 481
Shin, SangChul 332
Shin, Seongho 82
Shin, Soo-Young 491
Siddiqui, Faysal Adeem 170
Siradjev, Djakhongir 509
Soares, A.J.M. 441
Son, Jin hyuk 505
Song, Hyewon 200
Song, Wang-Cheol 383
Sugiyama, Keita 461
Sundaram, Shanmugham 221
Sur, Chul 273

Takami, Kazumasa 302
Takano, Chisa 461
Takano, Makoto 293
Talipov, Elmurod 574
Tan, Wayman 73
Tanaka, Yoshiaki 73, 92, 501
Tonouchi, Toshio 33
Tursunova, Shahnaza 221

Ueno, Hitoshi 451
Um, Tai-Won 403
Uppal, Amit 517

Wang, Yumei 23
Woo, Young-Wook 393

Yamada, Akiko 451
Yamamoto, Kimihiro 53
Yamamura, Tetsuya 210
Yanagimoto, Kiyoshi 293
Yang, Ok Sik 112
Yang, Soomi 530
Yeom, Heon Young 578
Yi, Myeongjae 232
Yim, Hong-bin 558
Yoo, Dae Seung 232
Yoo, Giljong 372, 566
Yoo, JaeHyoong 362
Yoo, Jae-Hyoung 550
Yoo, Sun-Mi 242
Yoon, Sangsik 63
Yoshida, Atsushi 431
Yoshihara, Kiyohito 352
Youn, Chan-Hyun 23, 200
Yun, Dong-Sik 312

Zhang, Lin 23
Zhanikeev, Marat 73, 92, 501
Zhu, Huamin 122, 142