Lecture Notes in Computer Science 2402
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Whie Chang (Ed.)

Advanced Internet Services and Applications

First International Workshop, AISA 2002
Seoul, Korea, August 1-2, 2002
Proceedings

Springer

Volume Editor

Whie Chang
CEO, C-EiSA, Co., Ltd.
135-090, JEIL Bldg. 4th, Samsung-Dong 168-26
Kangnam-Gu, Seoul, Korea
E-mail: wchang@c-eisa.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Advanced internet services and applications : first international workshop ;
proceedings / AISA 2002, Seoul, Korea, August 1 - 2, 2002. Whie Chang (ed.).
- Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ;
Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2402)
ISBN 3-540-43968-4

CR Subject Classification (1998): C.2, D.4, H.3, H.4, H.5.1

ISSN 0302-9743
ISBN 3-540-43968-4 Springer-Verlag Berlin Heidelberg New York

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

Printed in Germany

Typesetting: Camera-ready by author, data-conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10873625 06/3142 5 4 3 2 1 0

Preface

The rapid growth of the Internet and related services is changing the way we work, act, and even think in a manner that far exceeds the prediction set by field experts not long ago. It is now common belief that the Internet and its various applications covering both hardware appliances and software products will play an increasingly important role in everybody's daily lives. It is also our strong belief that the importance of the collaborative research and development efforts focusing on the Internet among academia, industry, and regulating government bodies cannot be overemphasized.

It is our great pleasure to hold the First International Workshop on Advanced Internet Services and Applications (AISA) 2002. The workshop is aimed to provide an international forum to share new ideas and research results in the area of the list of workshop topics. Under the main theme "Advances in Internet Services and Applications", the workshop topics include QoS architecture, reliability, security, web acceleration, reverse/proxy caching schemes, content delivery network, distributed/fault-tolerant architecture, storage/backup solutions, media and streaming server, switching technology, and home networking. We have solicited papers on these topics and attracted paper submissions from technically renowned organizations.

After an anonymous peer-review procedure, 28 papers that met the workshop scope and standard were finally selected for presentation from among more than 70 submitted papers. The high-quality of the AISA 2002 program would not have been possible without the dedicated cooperation of the organizing and program committees, and all the other field experts who served as reviewers. Our special thanks go to Keumchan Whang who performed a remarkable job as the Program Chair. We would also like to acknowledge the sponsorship of the Ministry of Commerce, Industry, and Energy (MOCIE) of Korea. The MOCIE has also been providing funding for the Development of High-Performance Scalable Web Servers as one of its primary government R&D projects, and we believe the technical excellence of the project and the financial support helped to raise the quality of this workshop to an even higher level. Leading companies, research institutes, and universities in the field of Internet technology, that presented at this workshop are surely the main technical contributors. A total of 12 associated companies, including Samsung Electronics and SsangYong Information and Communication, kindly sponsored this workshop. Finally, we wish to thank the Automation Technology Research Institute of Yonsei University in Korea for supporting this workshop and sharing its invaluable resources for the successful realization of this meaningful event.

August 2002 Whie Chang

Organization

The First International Workshop on Advanced Internet Services and Applications (AISA) 2002 was sponsored by MOCIE (Ministry of Commerce, Industry, and Energy) and Yonsei University, and held August 1–2, 2002, in Seoul, KOREA.

Executive Committee

Conference Chair:	Keumchan Whang (Yonsei University, KOREA)
Program Chair:	Whie Chang (C-EiSA, KOREA)
Workshops:	Keumchan Whang (Yonsei University)

Program Committee

Conference Chair:	Keumchan Whang (Yonsei University)
Program Chair:	Whie Chang (C-EiSA)
Organizing Chair:	Hagbae Kim (Yonsei University)
Tutorials:	Iltaek Lim (Serome Technology, KOREA)
Workshops:	Keumchan Whang (Yonsei University)
Panels:	Youngyong Kim (Yonsei University)
Exhibition:	Sangyong Hyun (SNS Solution, KOREA)

Referees

Allan L. White
Atsushi Fukuda
Bill Mathrani
Changmoo Ha
Chulhoon Lee
Chunhee Woo
Deogkyoo Lee
Euiseok Nam
Gordan Yang
Hangjae Yoo
Hansen Cai
Insuk Ahn
Jaehyeong Kim
Jaekeun Ahn
Jongjin Park
Joseph Hibey
Junjae Yoo
Lu Xianhe
Moonseol kang
Sangmine Lee
Sangyoub Kim
Seokho Jang
Seokkyu Kweon
Seunghwa Yoo
Seungyong Lee
Sinjun Kang
Sungho Kang
Sunghyun Choi
Wonjin Sung
Yaungjo Cho

Sponsoring Institutions

Samsung Electronics
SsangYong Information and Communications
Serome Technology
Hyundai Datanet
CVnet
C-EiSA
Mediaworks
SENA Technology
SNS Solution
SysGate
Seoul Commtech
WIZ Information Technology

Table of Contents

A New Survival Architecture for Network Processors*

Sungil Bae, Daesik Seo, Gilyoung Kang, and Sungho Kang

Dept. of Electrical and Electronic Engineering, Yonsei University
134 Sinchon-Dong, Seodaemun-Gu,
Seoul 120-749, Korea
shkang@yonsei.ac.kr

Abstract. We propose a new survival architecture that analyzes a network environment and decides the classification of the faults through the packet flow for network processors. The proposed architecture recognizes the unusual state in the network when the packet, which is the input of a network processor, can not be transmitted for a while. The network processor with the abnormality of the packet transmission detection generates a test packet so as to report the abnormality and to collect the information for the analysis of the source and the classification of the faults. According to this process, the proposed architecture derives the cause of the fault and the solution. Finally, the stability of the packet process is increased because the viability of the network processor is increased.

1 Introduction

A transmission capacity of the network is highly increased due to the sudden increase in the requirement for the network. In order to transmit a plenty of data, the researches and the developments of the high speed network devices are to be accelerated. In recent years, the network processor with a hardware-based packet processing structure is proposed to process the packets faster. To improve the performance of the packet process, they use the combination of two or more network processors [1-9].

The network processor is designed based on the simple processes of a network protocol while general processors perform the complex operations. Therefore, the constituent units of the network processor are relatively simple. And the multiple integration of the fundamental unit improves the packet processing power. Fig. 1 shows the structure of the conventional network processor composed of an interface unit, a memory, a packet processing unit, and a core. The network processor, in general, uses a high performance commercial core instead of ASIC (application specific integrated circuits), which is specified in the network applications.

* This work was supported by Ministry of Commerce, Industry and Energy, Republic of Korea.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 1–9, 2002.

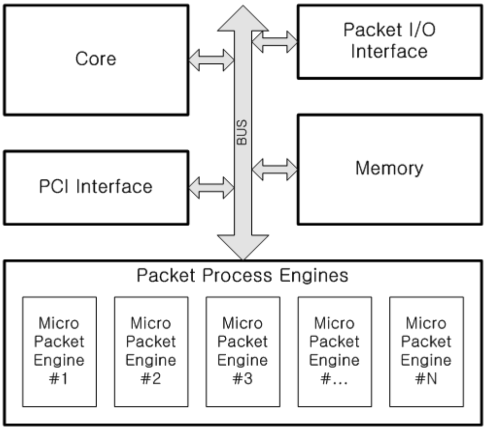


Fig. 1. Block diagram of the conventional network processor

In the typical design, the focus of the network processor design is the packet processing speed. However this increases the fault factors. In the weakness of the network processor, the faults inside the network processor is detectable using BIST and/or the boundary scan schemes [10,11]. But, it is impossible for the typical network processor to detect the faults caused outside the processor.

An EHW (evolvable hardware) is used to overcome these problems. The EHW using a biological method has an ability to settle the problems, which is expected in the practical applications [12-14].

We apply the survivability among the various EHW concepts to the network processor so as to acquire the solutions of the complex network problems. The survivability suggests the solution of the problems of the individual network processor and among the network processors. The survivability makes the network processor more flexible than the conventional ones. Thus, the network processor with the survival architecture performs the correct operations in the unexpected cases. It adapts the diverse network environments that are unpredictable.

2 Bio Inspired Hardware

An evolvable hardware is based on the idea of combining a reconfigurable hardware device with genetic algorithms to execute reconfiguration autonomously. The structure of reconfigurable hardware devices can be changed any number of times by downloading a bit string called configuration bits into the device. FPGA (field programmable gate array) and PLD (programmable logic devices) are typical examples of reconfigurable hardware devices.

Lately, the concept of the survival is expanded from a reconfigurable hardware to various applications, modeled by the biological characteristics. This concept is called BIH (bio inspired hardware) in this paper. A survival scheme is one of the expanded survival concepts, inspired from the group life of the living things,

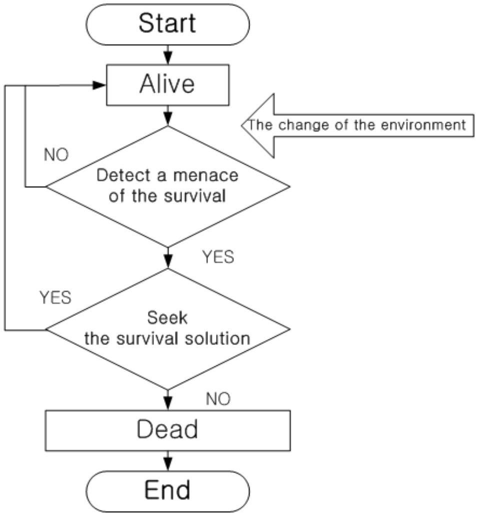

Start
Alive
The change of the environment
NO
Detect a menace of the survival
YES
YES
Seek the survival solution
NO
Dead
End

Fig. 2. The concept of the instinct for survival

and it is the same concept of the biological field. The survival scheme as shown in Fig. 2, is the behavior to survive in sudden environmental change similar to a creature that has an instinct for survival [15-18].

While the conventional architecture has the mechanical operations, the survival architecture analyzes the operating surroundings and derives the optimal solution. The general design scheme optimizes the operations for the expected tasks and failures. However, in an unexpected case, it can not derive the solution, and is replaced by a new design.

In order to solve the problem of the unexpected case, many researches have focused the survival. As the survival is one of the basic features of the creatures, the living organisms maintain their existences in their living environments. When the survival is applied to the hardware, the hardware supports the solution for the faults induced in the various operating environments. The components of survival for the hardware are assorted as follows. First of all, the recognition ability for current or changed environments is required. Namely, when the working hardware shows abnormal operations, the hardware must recognize the abnormality and prevent the spread of the damage. Secondly, the analysis on the recognized problem is required. The hardware decides whether the occurred problem is intrinsic or extrinsic, and should know the problem in detail. Thirdly, the hardware has to find the solution to the analyzed problem. If the problem is an internal one, then the hardware tests its components and collects the recovery information. If the hardware can not repair itself, then it reports the fault to the outside and requests an aid. Especially, in the co-operating system with several hardwares, the survival scheme protects the system operation. Since the third concept is the most difficult for hardware, a new algorithm, like a genetic algorithm, is applied [19].

3 Survival Technique for Network Processors

We applied the survival concept, as stated above to a network processor. The network processor is a high speed hardware that processes a large quantity of network packets. In practice, the network device has multiple network processors, it is necessary to analyze the operating environment for multiple network processors.

Therefore, we analyze the faults for the network processor in the environment of the multiple processors. At first, the network processor has a fault itself. This fault is caused by the component block failure, the interconnection failure between these blocks, the test controller failure, and the unpredictable case. But, many test schemes such as BIST and scan structure, can detect the fault effectively. Secondly, when the multiple network processors are used in the network system, many new problems may possibly arise. These problems are generated from the relationship with a neighborhood processor. But, the detection of these faults is very difficult. Because the network processor consists of simple units, there is no effective method that tests the relationship among the network processors. It is the only method to detect the new faults that we observe the flow of the incoming packets.

The packets in the normal state are transmitted from the network to the network device. The master processor of the network device distributes these packets to individual network processor via job scheduling. If there is no packet without the process idle signal for a long time, there are some problems between the master and the network processor, or between the network processors. The causes of this fault are diagnosed as follows. Firstly, the network processor that generates the incoming packets has a problem. Secondly, the network processor did not receive the idle signal from the master processor. In this case, the current packet processing has ended but the network processor can not detect an end. And the interconnection of the current processor is open. Thirdly, the main processor has a problem. At last, there is an unexpected fault (Fig. 3).

This paper proposes the new survival architecture for detection and treats the extrinsic problems mentioned above. The proposed architecture basically performs the self test when the power is on. If this test fails, then the fault is reported to the outside and the current processor gets disconnected from the whole packet process. In the case where the power-on test is passed, the processor does a normal operation. In the normal operation, if the transmitted packet is faulty or the transmission of the incoming packet does not continue for a while, the processor intuits that there are some problems. At this time, the network processor outputs the test packet, including the processor's ID (identification), and analyzes the fault. When the neighborhood processors and the main processor receive the test packet, they generate the information about the fault. They test themselves and output the results. When the current network processor receives the responses of the neighborhood processors and the main processor as the test results, they decide the characteristic of the fault through the analysis of the responses.

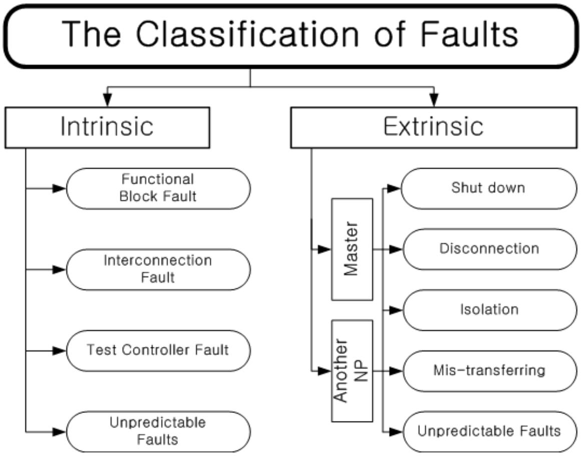


Fig. 3. The classification of faults in the practical application

When there are no responses, the current network processor may be isolated from the others (The whole interconnections of the current processor are not connected), or the output of the current processor may be open.

When there are some responses, we analyze the responses and conclude the fault type. If there is no response from the previous processor only, it is determined that the previous processor has the problem. If all the neighborhood network processors have no problems and there is no response from the main processor, the main processor is in an abnormal state.

When all responses are correct, we estimate that either the previous processor or the current processor has fault. Therefore, these processors perform the self test again.

Fig. 4 shows the flow diagram for the fault detection and the analysis schemes. Fig. 5 shows the program codes for the proposed survival scheme implemented in C language.

4 Proposed Survival Architecture

Fig. 6 shows that the proposed survival technique is implemented into the hardware with the network processor structure. In Fig. 6, a survival supervisor is the main unit to control the management of the fault detection, the fault analysis, and the fault recovery. The survival supervisor consists of a self test agent, a self test controller, a packet flow observer, a test packet generator, and a fault analyzer.

The self test agent is for the component units without the self test module. Every time the self test is required, this unit tests the component units and reports the test results to the fault analyzer. The agent has the self test structure and is composed of a test vector generator, a test response compactor, and a

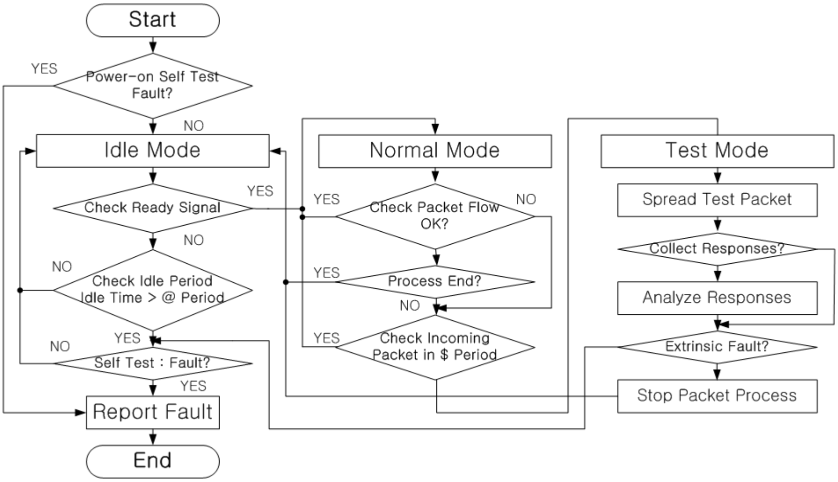
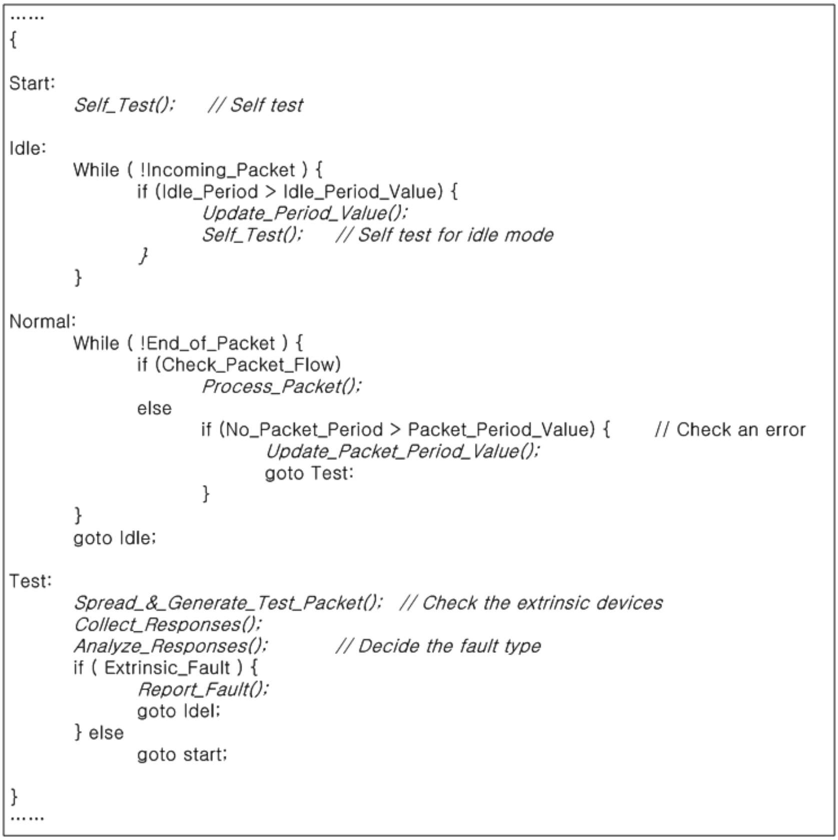


Fig. 4. Flow of the fault detecting and analyzing schemes



```
......
{

Start:
        Self_Test();     // Self test

Idle:
        While ( !Incoming_Packet ) {
                if (Idle_Period > Idle_Period_Value) {
                        Update_Period_Value();
                        Self_Test();     // Self test for idle mode
                }
        }

Normal:
        While ( !End_of_Packet ) {
                if (Check_Packet_Flow)
                        Process_Packet();
                else
                        if (No_Packet_Period > Packet_Period_Value) {       // Check an error
                                Update_Packet_Period_Value();
                                goto Test:
                        }
        }
        goto Idle;

Test:
        Spread_&_Generate_Test_Packet();   // Check the extrinsic devices
        Collect_Responses();
        Analyze_Responses();          // Decide the fault type
        if ( Extrinsic_Fault ) {
                Report_Fault();
                goto Idel;
        } else
                goto start;

}
......
```

Fig. 5. Proposed survival procedure

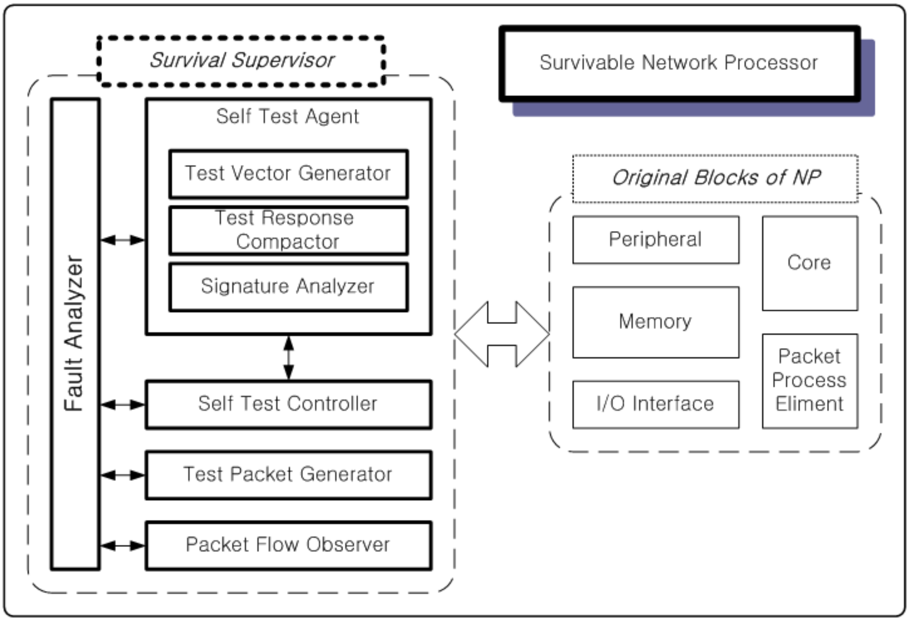


Fig. 6. Block diagram of the proposed network processor

signature analyzer similar to a BIST structure. The test vector generator makes the bit stream to test each unit. Hence, this unit can generate various lengths of test vectors.

The self test controller manages the self test units in the component units and the self test agent. When there is a request for the self test, the controller gives the command to perform the self test and takes the test results back. The controller collects the responses and forwards the arranged information to the fault analyzer. Besides, the controller regularly performs the self test when the network processor is in the idle mode. In addition to that, the number of times when the self test in the idle mode is changed according to the request frequency. The controller has the module to treat both the BIST controller for a BIST and the tap controller for a boundary scan.

The packet flow observer watches for the packets to come into the network processor. During the idle mode, the observer is inactive while the observer is active during the normal mode. In the normal mode, it checks the packet flow all the time. If there is no incoming packet for a specific period initiated at the beginning and the incoming packet is faulty, the controller detects an error in the outside and informs the error to the fault analyzer. The undetectable period of the packet is updated with a new value each time the abnormal flow is detected.

The test packet generator produces the unique packet including the special information when the packet flow observer detects the error. The test packet contains the processor's ID (identification) and the error information. This packet is useful for searching the processor with the problem.

Table 1. Function comparisons

Function Comparisons	The Conventional Network Processor	The Proposed Network Processor
Self Test	E(nable)	E
Self Test for Idle Mode	D(isable)	E
Interconnection Test	D	E
Networking Test	D	E
Master Test Request	D	E
Transfer Test	D	E

At last, the fault analyzer decides the fault types with various types of test information. In a specific case, the analyzer orders the network processor to stop the operation until the recovery signal is transmitted.

5 Performance Estimations

Table 1 shows the comparison of the proposed network processor with the conventional network processor. While the conventional network processor can perform only the self test, the proposed architecture can execute various tests, which are the self tests for idle mode; the interconnection test, the networking test, the master test request. The Self Tests for Idle Mode are the autonomous self test, performed when the network processor is in idle mode. The Interconnection Test knows whether the connectivity between the current processor and the neighborhood processor or the master processor is good or bad. The Networking Test can check the stability of the distributed packet process. When the Master Test Request is activated by the slave processor, the master must perform the self test operation. The Transfer Test can test the mis-transfer occurred by an irregular processing between the network processors.

Moreover, the proposed design can assort the faults in detail. As a result, the proposed architecture can not only prevent the processor-level failure, but also the system-level failure.

6 Conclusions

The network system using the typical network processor, which is able to do only the self test, can not forecast the system failure. But, the proposed architecture with the survival scheme can check the network and the neighborhood processors' stabilities. The survival architecture guarantees the stability of the distributed system and improves the accuracy of the packet process. Despite its additional survival ability, the proposed design does not affect the original structure of the network processor. Furthermore, the new survival architecture of a network processor can repair damaged packets and rehabilitates itself.

References

1. MENZILCIOGLU, O., etc: Nectar CAB : A high-speed network processor. Proceedings of IEEE International Joint Conference on Neural Networks (1991) 580–515
2. PALMER, R.P., etc: An architecture for appliation specific neural network processors. Computing and Control Engineering Journal (Dec. 1994) 260–264
3. XIAONING NIE, etc: A new network processor architecture for high-speed communications. IEEE Workshop on Signal Processing Systems (1999) 548–557
4. MEMIK, G., etc: NetBench: A benchmarking suite for network processors. Proceedings of IEEE/ACM International Conference on Computer Aided Design (2001) 39–42
5. WILLIAMS, J.:Architectures for network processing. Proceedings of Technical Papers on VLSI Technology, Systems, and Applications (2001) 61–64
6. KARIM, F., etc: On-chip communication architecture for OC-768 network processors. Proceedings of Design Automation Conference (2001) 678–683
7. DASGUPTA, S., etc: An enhancement to LSSD and some applications of LSSD in reliability, availability and serviceability. Proceedings of IEEE International Symposium on Fault-Tolerant Computing (1981) 32–34
8. GEPPERT, L.: The new chips on the block [network processors]—. IEEE Spectrum (Jan. 2001) 66-68
9. PAULIN, P.G., etc: Network processors: a perspective on market requirements, processor architectures and embedded S/W tools. Proceedings of Conference and Exhibition on Design, Automation and Test in Europe (2001) 420–427
10. AGRAWAL, V., etc: A Tutorial on Built-In Self-Test Part I:Principles. IEEE Design and Test of Computers (1993) 73–82 446–452
11. BUTT, H.H.:ASIC DFT techniques and benefits. Proceedings of IEEE ASIC Conference and Exhibition (1993) 46-53
12. Evolvable Systems: From Biology to Hardware, Lecture Notes in Computer Science 1259. Springer-Verlag (1997)
13. XIN YAO, etc: Promises and Challenges of Evolvable Hardware. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews (1999) 87–97
14. PATT, Y.:Requirements, Bottlenecks, and good fortune: agents for microprocessor evolution. Proceedings of the IEEE (2001) 1553–1559
15. NISBET, R., etc: The evolution of network-based corporate networks. Proceedings of Computer, Communication, Control and Power Engineering (1993) 471-474
16. CRINKS, R.A., etc: Evolution of network architectures. Proceedings of Military Communications Conference (2000) 1204-1208
17. WANG, Y.H., etc: Survivable wireless ATM network architecture. Proceedings of International Conference on Computer Communications and Networks (2000) 368–373
18. MORRISON, J.P., etc: Fault tolerance in the WebCom metacomputer. Proceedings of International Conference on Parallel Processing Workshops (2001) 245–250
19. GOLDBERG, D.: Genetic Algorithms in Search, Optimization, and Machine Learining. Addison Wesley (1998)

Mean Time to Unsafe Failure

Allan L. White

NASA Langley
Hampton, VA 23681-0001, USA
a.l.white@larc.nasa.gov

Abstract. With the increased use of electronics and networks comes a concern about their failure modes. If they fail, we wish they did so in a benign manner. Hence, a reliability parameter of interest is the expected time to a disastrous failure, where there is a loss of money, information, or time. The models and methods for computing this expectation, however, are still in an early stage. This paper presents new material that permits examining more complex and realistic models.

1 Introduction

If a large number of components operate for a long period of time, then there will almost certainly be failures. It is then important that the system fail in a safe manner. The meaning of safe depends on the circumstances. If it is a traffic control system, then safe means rerouting traffic or bringing traffic to a safe stop. For the internet it can mean rerouting the flow of data, not losing any data, or having a short downtime. For all such systems, a relevant parameter is the mean (or expected) time to an unsafe failure—mttuf.

This topic is new with a previous publication being [1] which considers just Markov models. This paper differs in four respects from [1]. It takes a simpler approach to computing Markov models. It includes the repair process in the computation. It uses a symbol manipulation program (The MATLAB Symbolic Math Toolbox) that permits handling larger and more realistic models. It derives a formula for mttuf from the single cycle parameters and applies this formula to a semi-Markov model.

One feature of mttuf computations is that the results are opposite of what we expect, even for the simple Markov models. Increasing redundancy decreases the mttuf. Increasing the component failure rate increases the mttuf. These issues still remain to be resolved. Perhaps continued investigation using even more realistic models will solve these problems. At the moment, we can only observe the strange behavior of systems with respect to mttuf.

Section two presents the method for computing the mttuf for Markov models. Section three derives the formula for mttuf in terms of single cycle parameters. Section four considers a semi-Markov model.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 11–20, 2002.

2 Markov Models

Mean time to unsafe failure for Markov models can be handled by the Laplace transform. Suppose f(t) is the density function for the time to unsafe failure. The Laplace transform derivations are

$$\begin{aligned} L[f(t)] &= \int_0^\infty e^{-st} f(t)dt \\ \frac{d(L[f(t)])}{ds} &= \int_0^\infty -te^{-st} f(t)dt \\ \lim_{s\to 0} \frac{d(L[f(t)])}{ds} &= -\int_0^\infty tf(t)dt \end{aligned} \tag{1}$$

Hence, the limit of the derivative of the Laplace transform equals the negative of the first moment. The Laplace transform can be obtained from the Chapman-Kolmogorov differential equations. Get the transform of each equation, then solve a set of linear algebraic equations.

2.1 Example: Threeplex with Coverage

A universal concept in reliability is that of coverage—the probability of a good outcome given a failure. One of the easiest ways to include coverage in a model is to multiply by a coverage factor (usually denoted by C), but this method has some limitations which are discussed in the next subsection. Consider a threeplex where the good outcome is a safe failure upon the failure of a component. The model, using a coverage factor, is given in Fig. 1. The component failure rate is λ; the coverage factor is C; the repair rate is ε; state 3 is fail safe; and state 4 is fail unsafe.

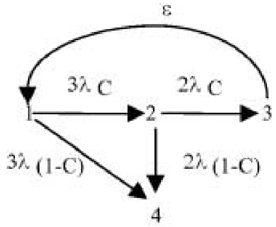


Fig. 1. Threeplex with coverage

The Chapman-Kolmogorov differential equations are

$$\begin{aligned} p_1'(t) &= -3\lambda p_1(t) + \epsilon p_3(t) \\ p_2'(t) &= 3\lambda C p_1(t) - 2\lambda p_2(t) \\ p_3'(t) &= 2\lambda C p_2 - \epsilon p_3(t) \\ p_4'(t) &= 3\lambda(1-C)p_1(t) + 2\lambda(1-C)p_2(t) \end{aligned} \tag{2}$$

Given the system begins in state 1, the Laplace transforms of the Chapman-Kolmogorov equations are

$$\begin{aligned} sP_1 - 1 &= -3\lambda P_1 + \epsilon P_3 \\ sP_2 &= 3\lambda C P_1 - 2\lambda P_2 \\ sP_3 &= 2\lambda C P_2 - \epsilon P_3 \\ sP_4 &= 3\lambda(1-C)P_1 + 2\lambda(1-C)P_2 \end{aligned} \tag{3}$$

where capital P_i is the Laplace transform of p_i (t). Solving for sP_4 yields

$$sP_4 = \frac{\lambda(2\lambda C^2 s - 2\lambda s + 2\epsilon\lambda C^2 - \epsilon s - s^2 + \epsilon C s - 2\lambda\epsilon + C s^2)}{6\lambda^2\epsilon C^2 - 6\lambda^2\epsilon - 5\lambda\epsilon s - 6\lambda^2 s - 5\lambda s^2 - \epsilon s^2 - s^3} \tag{4}$$

The step above is a point in the derivation where it is easy to solve for the wrong quantity. We want the Laplace transform of the density function, which is sP_4 , not the transform of the distribution function, which is P_4 .

Taking the negative of the derivative of sP_4 and substituting s=0 gives

$$\lim_{s\to 0}[sP_4]' = -3\frac{\lambda(2\lambda C^2 - 2\lambda - \epsilon - C\epsilon)}{(6\lambda^2\epsilon C^2 - 6\lambda^2\epsilon)} + 3\frac{\lambda(2\lambda\epsilon C^2 - 2\lambda\epsilon)(-6\lambda^2 - 5\lambda\epsilon)}{(6\lambda^2\epsilon C^2 - 6\lambda^2\epsilon)^2} \tag{5}$$

Suppose the parameter values are failure rate λ= 1e-4/hour repair rate ε = 1/(24 hours) = coverage C = 0.9999.

Substitution gives mttuf = 4.1786 e+7.

2.2 Example: Fourplex with Coverage

Suppose we have a threeplex with a design flaw. With probability P_F the system fails whenever a single component fails. The model for a threeplex with a design flaw is given in Fig. 2.

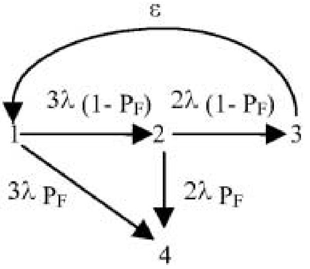


Fig. 2. Fourplex with Coverage

Fig. 2. is identical to figure with C = 1- P_F . Hence, using a coverage factor is equivalent to assuming a design flaw—the design is adequate for recovery

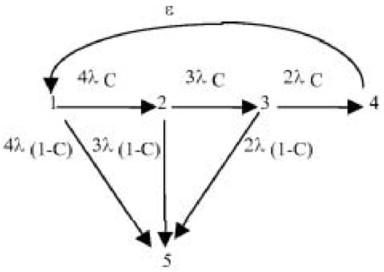


Fig. 3. Threeplex with Triplicated Monitor

from some, but not all, failures. A general principle is that redundancy does not overcome design flaws. In fact, redundancy can make matters worse. Consider a fourplex with a coverage whose model is given in Fig. 3.

With the same parameters as for the threeplex failure rate λ = 1e-4/hour repair rate $\varepsilon = 1/(24$ hours) = coverage C = 0.9999. The mttuf = 3.6190 2+7, which is less than the mttuf for the threeplex.

3 Threeplex and Fourplex with a Monitor

If lack of coverage is due to the failure of a critical number of components instead of a design flaw, then additional redundancy may increase reliability. Consider a threeplex with a triplicated monitor as shown in Fig. 4. This systems will fail safe if a majority of the monitors have not failed. The working components have failure rate λ = 1e-4/hour, and the monitor components have failure rate α = 1e-6/hour. The repair rate is ε =1/(24 hours).

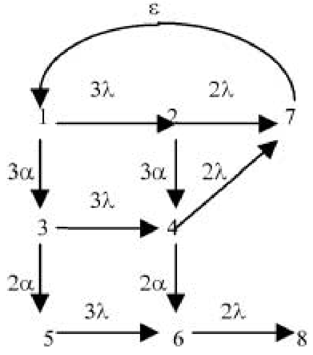


Fig. 4. Threeplex with Triplicated Monitor

For the threeplex with a triplicated monitor, mttuf = 2.7150 e+7 Consider a fourplex with a triplicated monitor as shown in Fig. 5.

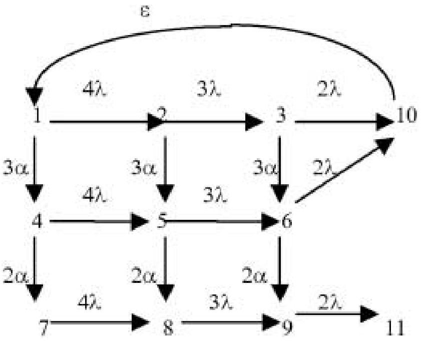


Fig. 5. Fourplex with Triplicated Monitor

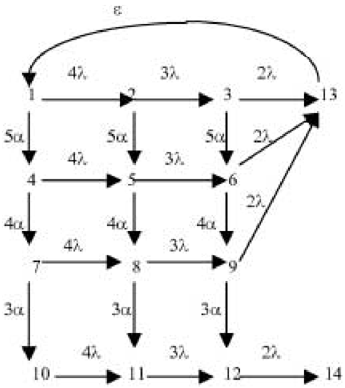


Fig. 6. Fourplex with Quintuple Monitor

For this system the mttuf = 2.3376 e+7, which is less than the threeplex with a monitor (2.7150 e+7)

A conjecture is that adding a component to the working group makes it more likely that the monitor will fail before the working group fails. Since the system fails unsafe if the monitor fails before the working group, the attempt to increase reliability by more redundancy yields a smaller mttuf. To test this conjecture, consider a working fourplex with a quintuple monitor, depicted in Fig. 6.

The fourplex with quintuple monitor has mttuf = 3.8707 e+8, which is an increase in mttuf compared to the previous systems.

3.1 A Failure Rate Example

It's possible for some computations to appear contradictory. For instance, it seems reasonable that increasing the failure rate will decrease the mttuf. The threeplex with $\lambda = 1$ e-2/hour, however, has mttuf = 3.3904 e+9. (Compared to a mttuf = 4.1786 e+7 for $\lambda = 1$ e-4/hour.) The conjecture for this example is the same as the conjecture above—to get a large mttuf it is better for the working components to fail before the monitor fails.

4 Mean Time to Unsafe Failure in Terms of Single Cycle Parameters

Computing the mttuf for Markov models is straightforward, although possibly tedious. Computing the mttuf for semi-Markov and globally time dependent models can be analytically hard. One method of reducing the burden is to break the computation into two parts. The first part considers the model without the repair transitions, which we will call the single cycle model. For instance, the single cycle model for a threeplex with coverage is given in Fig. 7.

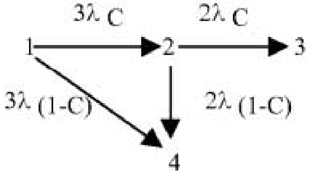


Fig. 7. Single Cycle Model of a Threeplex with Coverage

The presence of several absorbing states in the model above implies that their mean times must be conditioned. Strictly speaking, the mean time to state 3 is infinite since the system can go to state 4 and never arrive in state 3. Hence, we need the mean arrival time for state 3 given the system goes to state 3. Similarly for state 4.

The conditional mean can be handled by Laplace transforms for Markov models (and other models for which the Laplace can be computed). Suppose g(t) is a density function for an event. It can be defective (its integral is less than one). Its Laplace transform is

$$L[g(t)] = \int_0^\infty e^{-st} g(t)dt \tag{6}$$

The probability of the event is

$$\int_0^\infty g(t)dt = \lim_{s\to 0} L[g(t)] \tag{7}$$

Hence, the conditional mean is given by

$$\mu = \lim_{s\to 0} \frac{-\{[L[g(t)]]'\}}{L[g(t)]} \tag{8}$$

4.1 Derivation of the Single Cycle Formula for mttuf

For the first part of the computation, the probabilities and conditional mean times for safe failure and unsafe failure are computed. The second part uses the mean time for repair and the formula derived below to compute the mean-time-to-unsafe-failure. The derivation of the formula uses an intermediate result about summing a series.

$$\sum_{n=1}^{\infty}(n-1)a^{n-1} = \sum_{n=1}^{\infty} na^n = a\sum_{n=1}^{\infty}[a^n]'$$

Since a power series can be differentiated term by term,

$$= a\left[\sum_{n=1}^{\infty} a^n\right]' = a\left[-1 + \frac{1}{1-a}\right]' = \frac{a}{(1-a)^2} \tag{9}$$

For a single cycle let the probability of fail safe be P_S , the conditional mean time to fail safe be M_S , the conditional mean time to fail unsafe be M_U , and the mean time for repair be M_R . The mean time to unsafe failure is

$$\sum_{n=1}^{\infty}[(n-1)M_S + (n-1)M_R + M_U]P_S^{n-1}(1-P_S)$$
$$= (M_S + M_R)\left[\frac{P_S}{(1-P_S)}\right] + M_U \tag{10}$$

In the infinite series, the term in brackets is the mean time to unsafe failure given (n-1) safe failures followed by (n-1) repairs. This is multiplied by the probability of (n-1) safe failures followed by an unsafe failure. The series is summed using equation (9) above.

4.2 Example

We will recompute the mttuf for the first example using the single cycle formula. the complete model is given in figure 1. The single cycle model is depicted above in Fig. 7. The Laplace transform equations for the model in Fig. 7. are

$$\begin{aligned} sP_1 - 1 &= -3\lambda P_1 \\ sP_2 &= 3\lambda C P_1 - 2\lambda P_2 \\ sP_3 &= 2\lambda C P_2 \\ sP_4 &= 3\lambda(1-C)P_1 + 2\lambda(1-C)P_2 \end{aligned} \tag{11}$$

The Laplace transforms of the (defective) density functions, the negative of their derivatives, the probabilities, and the conditioned means are

$$sP_3 = \frac{6\lambda^2 C^2}{(s+3\lambda)(s+2\lambda)}$$

$$\begin{aligned}
sP_4 &= \frac{3\lambda(1-C)}{(s+3\lambda)} + \frac{6\lambda^2 C(1-C)}{(s+3\lambda)(s+2\lambda)} \\
-[sP_3]' &= 6\lambda^2 C^2 \left[\frac{1}{(s+3\lambda)^2(s+2\lambda)} + \frac{1}{(s+3\lambda)(s+2\lambda)^2}\right] \\
-[sP_4]' &= 3\lambda(1-C)\left[\frac{1}{(s+3\lambda)^2}\right] \\
&\quad +6\lambda^2 C(1-C)\left[\frac{1}{(s+3\lambda)^2(s+2\lambda)} + \frac{1}{(s+3\lambda)(s+2\lambda)^2}\right] \\
Prob\{state3\} &= \lim_{s\to 0} sP_3 = C^2 \\
Prob\{state4\} &= \lim_{s\to 0} sP_4 = 1 - C^2 \\
\mu_3 &= \lim_{s\to 0} \frac{-[sP_3]'}{sP_3} = \frac{1}{3\lambda} + \frac{1}{2\lambda} \qquad (12) \\
\mu_4 &= \lim_{s\to 0} \frac{-[sP_4]'}{sP_4} = \frac{1}{1+C}\frac{1}{3\lambda} + \frac{C}{1+C}\left[\frac{1}{3\lambda} + \frac{1}{2\lambda}\right]
\end{aligned}$$

For the model in figure 7, the probability of failing safe is P$_S$ = C^2 . The mean time to repair is M$_R$ = $1/\varepsilon$. Hence, the single cycle formula gives

$$\begin{aligned}
MTTUF &= (M_S + M_R)\left[\frac{P_S}{(1-P_S)}\right] + M_U \\
&= \left(\frac{5}{6\lambda} + \frac{1}{\epsilon}\right)\left(\frac{C^2}{1-C^2}\right) + \frac{1}{1+C}\frac{1}{3\lambda} + \frac{C}{1+C}\left[\frac{1}{3\lambda} + \frac{1}{2\lambda}\right] \\
&= \frac{C^2}{\epsilon(1-C^2)} = \frac{2+3C}{6\lambda(1-C^2)} \qquad (13)
\end{aligned}$$

The simplification of the result for the entire model (formula (5)) yields

$$\begin{aligned}
MTTUF &= -3\frac{\lambda(2\lambda C^2 - 2\lambda - \epsilon - C\epsilon)}{(6\lambda^2\epsilon C^2 - 6\lambda^2\epsilon)} + 3\frac{\lambda(2\lambda\epsilon C^2 - 2\lambda\epsilon)(-6\lambda^2 - 5\lambda\epsilon)}{(6\lambda^2\epsilon C^2 - 6\lambda^2\epsilon)^2} \\
&= \frac{-6\lambda^2(C^2-1)}{6\lambda^2\epsilon(C^2-1)} + \frac{3\lambda\epsilon(1-C)}{6\lambda^2\epsilon(C^2-1)} + \frac{-6\lambda^3\epsilon(C^2-1)(6\lambda+5\epsilon)}{36\lambda^4\epsilon^2(C^2-1)} \\
&= -\frac{1}{\epsilon} - \frac{1}{2\lambda(1+C)} + \frac{1}{\epsilon(1-C^2)} + \frac{5}{6\lambda(1-C^2)} \\
&= \frac{C^2}{\epsilon(1-C^2)} + \frac{2+3C}{6\lambda(1-C^2)} \qquad (14)
\end{aligned}$$

The two approaches give the same answer (which is reassuring).

5 A Semi-Markov Example

A system consists of a working unit plus a cold spare. When the working unit fails, with probability C it activates the spare and sends a signal that it has failed.

Upon receiving the signal, a repair crew is sent to the site where they replace both the failed unit and the operating spare. If the operating spare fails before the repair crew arrives, it can fail safe (probability C) or unsafe (probability 1-C). If it fails safe, it sends a signal and traffic is either halted or rerouted. The model for this system is given in Fig. 8. where f(t) is the failure density function and h(t) the repair density function.

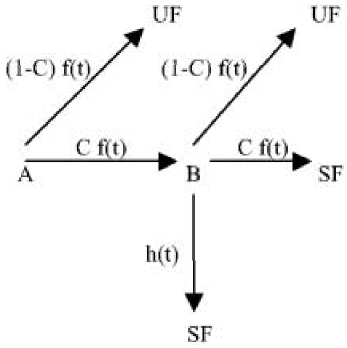


Fig. 8. Semi-Markov for a System with Coverage,a Cold Spare,and Repair

The single cycle parameters (in unreduced forms that displays the conditioning) are

$$
\begin{aligned}
p_1 &= \int_0^\infty f(t)[1 - H(t)] \qquad p_2 = \int_0^\infty h(t)[1 - F(t)] \\
P\{SF\} &= C^2 p_1 + C p_2 \qquad P\{UF\} = (1 - C) + C(1 - C)p_1 \\
\mu(SF) &= \frac{C^2 p_1 \left[\frac{\int_0^\infty tCf(t)}{C} + \frac{\int_0^\infty tCf(t)[1-H(t)]}{Cp_1}\right]}{P\{SF\}} \\
&+ \frac{Cp_2 \left[\frac{\int_0^\infty tCf(t)}{C} + \frac{\int_0^\infty th(t)[1-F(t)]}{p_2}\right]}{P\{SF\}} \\
\mu(UF) &= \frac{(1 - C)\left[\frac{\int_0^\infty t(1-C)f(t)}{(1-C)}\right]}{P\{UF\}} \\
&+ \frac{C(1 - C)p_1 \left[\frac{\int_0^\infty tCf(t)}{C} + \frac{\int_0^\infty t(1-C)f(t)[1-H(t)]}{(1-C)p_1}\right]}{P\{UF\}}
\end{aligned}
\qquad (15)
$$

The mean time to repair requires some derivation. If the repair crew arrives before the second component fails, then the repair time is negligible. If the second component fails safe before the repair crew arrives, then the average repair time is the difference of two (conditional) averages: the average arrival time of the crew given it arrives after the second component fails minus the average failure time of the second component given it fails safe before the crew arrives. This difference is

$$diffmean = \frac{\int_0^\infty th(t)F(t)}{\int_0^\infty h(t)F(t)} - \frac{\int_0^\infty tCf(t)[1 - H(t)]}{Cp_1} \tag{16}$$

The overall average repair time is

$$M_R = \frac{p_2(0) + p_1(diffmean)}{p_2 + Cp_1} \tag{17}$$

Suppose component failure is a normal distribution with mean = 0.5 years and standard deviation = 0.1 years. Suppose repair is normal with mean = 0.02 years and standard deviation = 0.01 years. Then all the above plus the formula in section three yields mttuf = 5.18 years.

6 Summary

Mean time to unsafe failure is a new topic, but it is applicable in many different fields. This paper offers both theoretical and computational extensions to the state of the art. The computation for Markov models has been simplified and now includes system repair time. A formula for mttuf in terms of single cycle parameters makes it possible to examine semi-Markov models. Unfortunately, there was not room in this paper to include a globally time dependent example. More remains to be done. In particular, the current results are often the opposite of what we expect. Perhaps some automatic model generation and computation programs will make it possible to construct and compute more realistic models.

References

1. Choi, Wang, and Trivedi: Conditional MTTF and its Computation in Markov Reliability Models, Proceeding 1993 Reliability and Maintainability Symposium, (1993) 55–63.

Performance of Antenna Arrays with Reverse-Link Synchronous Transmission Technique for DS-CDMA System in Multipath Fading Channels

Youngseok Kim[1], Seunghoon Hwang[2],
Dongkyoon Cho[1], and Keumchan Whang[1]

[1] The Department of Electrical and Electronic Engineering,
Yonsei University, Seoul, Korea
kcwhang@yonsei.ac.kr
[2] The Standardization and System Research Gr., UMTS System Research Lab.,
LG Electronics, Inc., Kyungki-Do, Korea

Abstract. This paper introduces an improved antenna arrays (AA) in which the reverse-link synchronous transmission technique (RLSTT) is employed to improve the estimation of covariance matrices at Beamformer-RAKE. By beamforming, the desired user's signal is enhanced and the co-channel interference (CCI) from other directions is reduced. For in-beam multiple access interference (MAI) reduction, the RLSTT is used in the first RAKE finger. The RLSTT is a robust approach that takes into account the fact that the in-beam MAI statistics at the beamformer may be less valid and lets it be more reliable. Theoretical analysis demonstrates that an increase in system capacity as a function of the number of antennas and the number of interferences.

1 Introduction

DS-CDMA systems exhibit a user capacity limit in the sense that there exist a maximum number of users that can simultaneously communicate over multipath fading channels for a specified level of performance per user. This limitation is caused by the co-channel interference (CCI), which includes both multiple access interference (MAI) between the multiusers and intersymbol interference (ISI) arising from the existence of different transmission paths.

An approach to increase the system capacity is the use of the spatial processing with antenna arrays (AA) at base station (BS)[1]-[3]. Generally, AA system consists of spatially distributed antennas and beamformer that generates a weight vector to combine the array output. Several algorithms have been proposed in the spatial signal processing to design the beamformers. A new space-time processing framework for adaptive beamforming with antenna arrays in CDMA was proposed [2]. This system uses a code-filtering approach in each receiving antenna to estimate both the channel vector and the optimum beamformer weights. For terrestrial mobile systems, the reverse-link synchronous

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 21–31, 2002.

transmission technique (RLSTT) has been proposed to reduce inter-channel interference over a reverse link [4]. In the RLSTT, the synchronous transmission in the reverse link can be achieved by controlling the transmission time continuously in each Mobile Station (MS). In a similar manner with the closed loop power control technique, the BS computes the time difference between the reference time that is generated in the BS and the arrival time of the dominant signal from each MS, and then transmits timing control bits, which order MSs to "advance or delay" their transmission times. The DS-CDMA system considered uses an orthogonal revere-link spreading sequence and the timing control algorithm that allows the main paths to be synchronized.

In this study, we propose and analyze an improved AA structure based on the RLSTT at BS. In this situation, the RLSTT is employed to reduce the in-beam MAI and improve the estimation of covariance matrices, and thus to further increase the system capacity. Therefore, the RLSTT is suitable for signal estimation at beamformer. Theoretical results show the performance of the proposed RLSTT at AA system.

2 Channel and System Model

We consider modulated BPSK DS-CDMA system over multipath fading channel. Assuming K active user ($k = 1, 2, \cdots, K$), the low-pass equivalent signal transmitted by user k is given by

$$s^{(k)}(t) = \sqrt{2P_k} b^{(k)}(t) g^{(k)}\left(t\right) a(t) \cos\left[\omega_c t + \phi^{(k)}\right] \tag{1}$$

where $a(t)$ is a pseudonoise(PN) randomization sequence, which is common to all the channels in a cell to maintain the CDMA orthogonality. $g^{(k)}\left(t\right)$ is the orthogonal channelization sequence and $b^{(k)}(t)$ is the user k's data waveform. In (1), P_k is the average transmitted power of k-th user, ω_c is the common carrier frequency, and $\phi^{(k)}$ is the phase angle of the k-th modulator to be uniformly distributed in $[0, 2\pi)$. The Walsh chip duration T_g and the PN chip interval T_c is related to the data bit interval T through the processing gain $N = T/T_c$. The complex lowpass impulse response of the vector channel for the k-th user may be written as [3]

$$\mathbf{h}_k(\tau) = \sum_{l=0}^{L^{(k)}-1} \beta_l^{(k)} e^{j\varphi_l^{(k)}} \mathbf{V}(\theta_l^{(k)}) \delta[\tau - \tau_l^{(k)}] \tag{2}$$

where $\beta_l^{(k)}$ is the Rayleigh fading strength and phase shift $\varphi_l^{(k)}$, and propagation delay $\tau_l^{(k)}$. $\mathbf{V}(\theta_l^{(k)})$ is the k-th user on the l-th path array response vector which is given by

$$\mathbf{V}(\theta_l^{(k)}) = [1 \quad e^{-j2\pi d \cos\theta_l^{(k)}/\lambda} \quad e^{-j4\pi d \cos\theta_l^{(k)}/\lambda} \cdots e^{-j2(M-1)\pi d \cos\theta_l^{(k)}/\lambda}]^T \tag{3}$$

We assume that all signals from the MS arrive at the BS antenna array uniformly of the mean angle of arrival(AOA) $\theta_l^{(k)}$ which is uniformly distributed in $[0, \pi)$. Assuming Rayleigh fading, the probability density function(pdf) of signal strength of the k-th user on the l-th propagation path,$l = 0, 1, \cdots, L^{(k)} - 1$, is given by

$$p(\beta_l^{(k)}) = \frac{2\beta_l^{(k)}}{\Omega_l^{(k)}} \exp\left(-\frac{\left(\beta_l^{(k)}\right)^2}{\Omega_l^{(k)}} \right) \tag{4}$$

The parameter $\Omega_l^{(k)}$ is the second moment of $\beta_l^{(k)}$ with $\sum\limits_{l=0}^{\infty} \Omega_l = 1$, and we assume it to be related to the second moment of the initial path strength $\Omega_0^{(k)}$, for exponenetial multipath intensity profile (MIP),by

$$\Omega_l^{(k)} = \Omega_0^{(k)} e^{-l\delta}, \ for \ 0 < l \leq L^{(k)} - 1, \ \delta \geq 0 \tag{5}$$

The parameter δ reflects the rate at which the decay of average path strength as a function of path delay occurs. Note that more realistic profile model may be the exponential MIP.

The receiver is a coherent RAKE with AA, where the number of fingers L_r is a variable less than or equal to $L^{(k)}$ which is the resolvable propagation paths with the k-th user. Perfect power control is assumed. The finger weights and phases are assumed to be perfect estimates of the channel parameters. The complex received signal is given as

$$\begin{aligned} \mathbf{r}(t) = \sqrt{2P} \sum_{k=1}^{K} \sum_{l=0}^{L^{(k)}-1} & \beta_l^{(k)} \mathbf{V}(\theta_l^{(k)}) b^{(k)} \left(t - \tau_l^{(k)}\right) \\ & \cdot g^{(k)} \left(t - \tau_l^{(k)}\right) a \left(t - \tau_l^{(k)}\right) \cos\left[\omega_c t + \psi_l^{(k)}\right] + \mathbf{n}(t) \end{aligned} \tag{6}$$

where $\psi_l^{(k)}$ is the phase of the l-th path of the k-th carrier. $\mathbf{n}(t)$ is the $M \times 1$ spatially and temporally white Gaussian noise Vector with zero mean and covariance which is given by

$$E\left\{\mathbf{n}\left(t\right)\mathbf{n}^H\left(t\right)\right\} = \sigma_n^2 \mathbf{I}$$

where, $\mathbf{I}$ is the $M \times M$ identity matrix and σ_n^2 is the antenna noise variance with $\eta_0/2$ and superscript H denotes the Hermitian-transpose operator.

The received signal is matched to the reference user's code. The l-th multipath matched filter output for the interest user $(k = 1)$ can be expressed as

$$\begin{aligned} \mathbf{y}_l^{(1)} &= \int_{\tau_l^{(1)}}^{\tau_l^{(1)}+T} \mathbf{r}(t) \cdot g^{(1)}(t - \tau_l^{(1)}) a(t - \tau_l^{(1)}) \cos[\omega_c t + \psi_l^{(1)}] dt \\ &= \mathbf{S}_l^{(1)} + \mathbf{I}_{l,mai}^{(1)} + \mathbf{I}_{l,si}^{(1)} + \mathbf{I}_{l,ni}^{(1)} \end{aligned} \tag{7}$$

Throughout this paper, we assume that the array geometry, which is the parameter of the antenna aperture gain, is a uniform linear array(ULA) of M identical

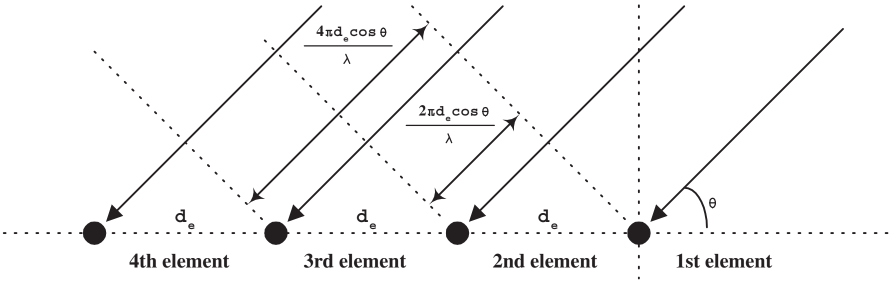

$\frac{4\pi d_e \cos\theta}{\lambda}$
$\frac{2\pi d_e \cos\theta}{\lambda}$
θ
d_e
d_e
d_e
4th element
3rd element
2nd element
1st element

Fig. 1. Antenna Array Model Geometry

sensors as illustrated in Figure 1. When a reference signal is not available, a common criterion for optimizing the weight vectors is the maximize the signal-to-interference plus noise ratio(SINR). In (7), $\mathbf{u}_l^{(1)} = \mathbf{I}_{l,si}^{(1)} + \mathbf{I}_{l,mai}^{(1)} + \mathbf{I}_{l,ni}^{(1)}$ is the total interference plus noise for the l-th path of the interest user. By solving the following problem, we can obtain the optimal weights to maximize the SINR.

$$\mathbf{W}_{l(opt)}^{(1)} = \max_{\mathbf{W}\neq\mathbf{0}} \frac{{\mathbf{W}_l^{(1)}}^H \mathbf{R}_{yy} \mathbf{W}_l^{(1)}}{{\mathbf{W}_l^{(1)}}^H \mathbf{R}_{uu} \mathbf{W}_l^{(1)}}$$

where, $\mathbf{R}_{yy}$ and $\mathbf{R}_{uu}$ are the second-order correlation matrices of the received signal subspace and the interference plus noise subspace, respectively. The solution is corresponded to the largest eigenvalue of the generalized eigenvalue problem in matrix pair ($\mathbf{R}_{yy}$,$\mathbf{R}_{uu}$). Therefore, we can obtain the maximum SINR when the weight vector $\mathbf{W}_{l(opt)}^{(1)}$ equals the principle eigenvector of the matrix pair, which is given by

$$\mathbf{R}_{yy} \cdot \mathbf{W}_{l(opt)}^{(1)} = \lambda_{\max} \cdot \mathbf{R}_{uu} \cdot \mathbf{W}_{l(opt)}^{(1)} \tag{8}$$

From (7) and (8), the corresponding beamformer output for the l-th path of the interest user is

$$\hat{z}_l^{(1)} = {\mathbf{W}_l^{(1)}}^H \cdot \mathbf{y}_l^{(1)} = \hat{S}_l^{(1)} + \hat{I}_{l,mai}^{(1)} + \hat{I}_{l,si}^{(1)} + \hat{I}_{l,ni}^{(1)} \tag{9}$$

where

$$\hat{S}_l^{(1)} = \sqrt{P/2}\beta_l^{(1)} C_{ll}^{(1,1)} b_0^{(1)} T$$

$$\begin{aligned}\hat{I}_{l,mai}^{(1)} = \sqrt{P/2} \sum_{k=2}^{K} \sum_{j=0}^{L^{(k)}-1} \beta_j^{(k)} C_{lj}^{(1,k)} \\ \times \left\{ b_{-1}^{(k)} RW_{k1} \left[\tau_{lj}^{(k)}\right] + b_0^{(k)} \widehat{RW}_{k1} \left[\tau_{lj}^{(k)}\right] \right\} cos \left[\psi_{lj}^{(k)}\right]\end{aligned}$$

$$\hat{I}_{l,si}^{(1)} = \sqrt{P/2} \sum_{\substack{j=0 \\ j\neq l}}^{L^{(1)}-1} \beta_j^{(1)} C_{lj}^{(1,1)} \left\{ b_{-1}^{(1)} RW_{11} \left[\tau_{lj}^{(1)}\right] + b_0^{(1)} \widehat{RW}_{11} \left[\tau_{lj}^{(1)}\right] \right\} \cos \left[\psi_{lj}^{(1)}\right]$$

$$\hat{I}_{l,ni}^{(1)} = \int_{\tau_l^{(1)}}^{\tau_l^{(1)}+T} \mathbf{W}_l^{(1)H} \cdot \mathbf{n}(t) g^{(1)}(t-\tau_l^{(1)}) a(t-\tau_l^{(1)}) \cos[\omega_c t + \psi_l^{(1)}]dt$$

with $b_0^{(1)}$ being the information bit to be detected, $b_{-1}^{(1)}$ is the preceding bit, $\tau_{lj}^{(k)} = \tau_j^{(k)} - \tau_l^{(1)}$ and $\psi_{lj}^{(k)} = \psi_j^{(k)} - \psi_l^{(1)}$. $C_{lj}^{(1,k)} = \mathbf{W}_l^{(1)H} \cdot \mathbf{V}\left(\theta_j^{(k)}\right)$ represents the spatial correlation between the array response vector of the k-th user at the j-th multipath and the weight vector of the interest user at the l-th multipath, and $\mathbf{W}_l^{(1)} = \left[w_{l,1}^{(1)}\ w_{l,2}^{(1)}\ \cdots\ w_{l,M}^{(1)}\right]^T$ is the $M \times 1$ weight vector for the l-th path of the first user. RW and $\widehat{RW}$ are Walsh-PN continuous partial cross-correlation functions defined by $RW_{k1}(\tau) = \int_0^{\tau} g^{(k)}(t-\tau)a(t-\tau)\ g^{(1)}(t)a(t)dt$ and $\widehat{RW}_{k1}(\tau) = \int_{\tau}^{T} g^{(k)}(t-\tau)a(t-\tau)g^{(1)}(t)a(t)dt$. From (9), we see that the outputs of the l-th branch, $l = 0, 1, \cdots, L_r - 1$, consists of four terms. The first term represents the desired signal component to be detected. The second term represents the MAI from the $(K-1)$ other simultaneous users in the system. The third term is the self-interference (SI) for the reference user. Finally, the last term is the Gaussian random variable due to the AWGN process.

3 AA with RLSTT

In our analysis, the evaluation is carried out for the case in which the arrival time of paths is modeled as asynchronous in every branch (i.e., for multipaths) but as synchronous in the first branch (i.e., for main paths) exceptionally. We model the MAI terms of the first branch and the other branches as Gaussian process with variances equal to the MAI variances for $l = 0$ and for $l \geq 1$, respectively. Here for the sake of simplicity, we assume that the sensor spacing is the half of the carrier wavelength and the number of the element is four($M = 4$). Using the derived results in [4], the variance of MAI for $l = 0$, conditioned on $\beta_l^{(1)}$ and λ_k is

$$\sigma_{mai,0}^2 = \frac{2E_b T\,(2N-3)}{15N\,(N-1)\cdot\left\{\sigma_{s,0}^{(1)}\right\}^4} \left\{\beta_0^{(1)}\right\}^2 \sum_{k=2}^{K} \sum_{j=1}^{L^{(k)}-1} \Omega_j^{(k)} \tag{10}$$

Similarly, the variance of MAI for $l \geq 1$ is

$$\sigma_{mai,l}^2 = \frac{4E_b T\,(N-1)}{15N^2\cdot\left\{\sigma_{s,l}^{(1)}\right\}^4} \left\{\beta_l^{(1)}\right\}^2 \sum_{k=2}^{K} \sum_{j=0}^{L^{(k)}-1} \Omega_j^{(k)} \tag{11}$$

where $E_b = PT$ is the signal energy per bit and $\left\{\sigma_{s,l}^{(k)}\right\}^2$ is the undesired signal power of the l-th multipath. In [2], if the code length is large and if the total number of path is large, then the total interference-plus-noise can be modeled as a spatially white Gaussian random vector $\mathbf{R}_{uu,l}^{(k)} = \left\{\sigma_{s,l}^{(k)}\right\}^2 \mathbf{I}$, which will be evaluated in appendix. The conditional variance of $\sigma_{si,l}^2$ is approximated by [5]

$$\sigma_{si,l}^2 \approx \frac{2E_bT}{5N\left\{\sigma_{s,l}^{(1)}\right\}^4}\left\{\beta_l^{(1)}\right\}^2\sum_{j=1}^{L^{(1)}-1}\Omega_j^{(1)} \tag{12}$$

The variance of the AWGN noise term, conditioned on $\beta_l^{(1)}$, is given as

$$\sigma_{ni,l}^2 = \frac{4T\eta_0}{\left\{\sigma_{s,l}^{(1)}\right\}^4}\left\{\beta_l^{(1)}\right\}^2 \tag{13}$$

Accordingly, the output of the receiver is a Gaussian random process with mean

$$U_s = \sqrt{8E_bT}\sum_{l=0}^{L_r-1}\frac{\left\{\beta_l^{(1)}\right\}^2}{\left\{\sigma_{s,l}^{(1)}\right\}^2}$$

and the total variance equal to the sum of the variance of all the interference and noise terms. At the output of the receiver, the received SNR may be written in compact form as γ_s

$$\begin{aligned}
\gamma_s = \Bigg\{ & \frac{(2N-3)(K-1)\{q(L_r,\delta)-1\}}{60N(N-1)} \\
& \cdot\frac{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^4}{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^2+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^2} \\
& +\frac{(N-1)(K-1)q(L_r,\delta)}{30N^2}\cdot\frac{\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^4}{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^2+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^2} \\
& +\frac{\{q(L_r,\delta)-1\}}{20N}\cdot\frac{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^4+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^4}{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^2+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^2} \\
& +\frac{\eta_0}{2E_b\Omega_0}\cdot\frac{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^4+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^4}{\left\{\beta_0^{(1)}\right\}^2/\{\sigma_{s,0}\}^2+\sum_{l=1}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_s\}^2}\Bigg\}^{-1} \\
& \times\frac{1}{\Omega_0}\sum_{l=0}^{L_r-1}\left\{\beta_l^{(1)}\right\}^2/\{\sigma_{s,l}\}^2
\end{aligned} \tag{14}$$

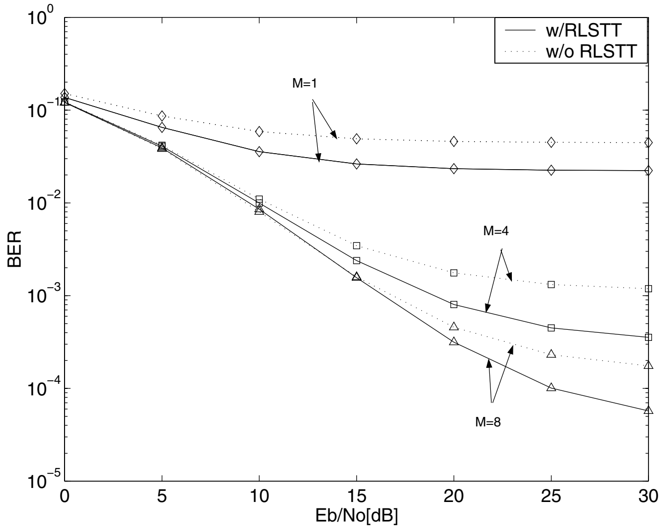


Fig. 2. AA w/ RLSTT vs. AA w/o RLSTT(User=72, δ=1.0, $L_r = L^{(k)} = 2$)

where $q\left(L_r, \delta\right) = \sum\limits_{l=0}^{L_r-1} e^{-l\delta} = 1 - e^{-L_r\delta}/1 - e^{-\delta}$, $\Omega_0^{(k)} = \Omega_0$ and $\sigma_{s,l}^{(k)} = \sigma_s$ for $l \ /= 0$. Since we assume that the multiple access interference is Gaussian, the bit error probability is a function of the SNR. Therefore, the average bit error probability P_e can be evaluated in a similar manner to [4].

4 Numerical Results

In all evaluations, processing gain is assumed to be 128, and the number of paths and taps in RAKE is assumed to be the same for all users and denoted two. In addition, the sensor spacing is the half of the carrier wavelength. The performance of AA with RLSTT and AA without RLSTT are compared with the different parameters such as decay factor and the number of antennas. Figures 2 and 3 show uncoded BER performance as a function of Eb/No and the number of users in the exponential MIP (decay factor=1.0). Note that the exponential MIP may give a more realistic profile model. The BER curves for $K = 72$ and the number of antenna, M = 1, 4 and 8 are plotted in fig. 2 and the BER curves for Eb/No=20dB are shown in fig.3. Using AA together with the RLSTT enhances the performance of the demodulated data and thus somewhat removes the error floor at high SNR. Although the performance gains between two schemes decrease as the number of antenna increases, AA with RLSTT shows significant performance gain when the number of users increases. At a BER of 0.001, AA without RLSTT supports 48 users, while using RLSTT together with AA supports 84 users in case of using four antennas. In figs. 4 and 5, the performance of AA with RLSTT and AA without RLSTT are compared especially in

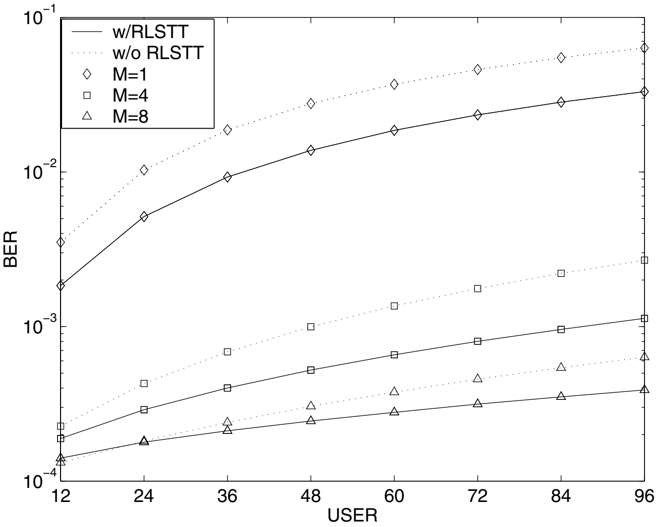


Fig. 3. AA w/ RLSTT vs. AA w/o RLSTT(δ=1.0, Eb/No=20[dB], $L_r = L^{(k)} = 2$)

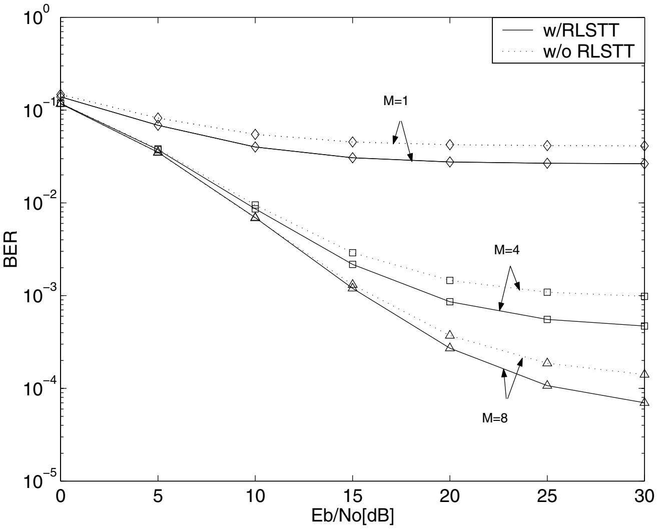


Fig. 4. AA w/ RLSTT vs. AA w/o RLSTT(User=72, δ=0.2, $L_r = L^{(k)} = 2$)

$\delta = 0.2$. In such a case, the performance gain is still kept even if the differences decrease. This results are based on the point that the RLSTT can offer better performance than conventional asynchronous transmission, especially for an exponentially decaying multipath intensity profile (MIP) with a large decay factor in [4].

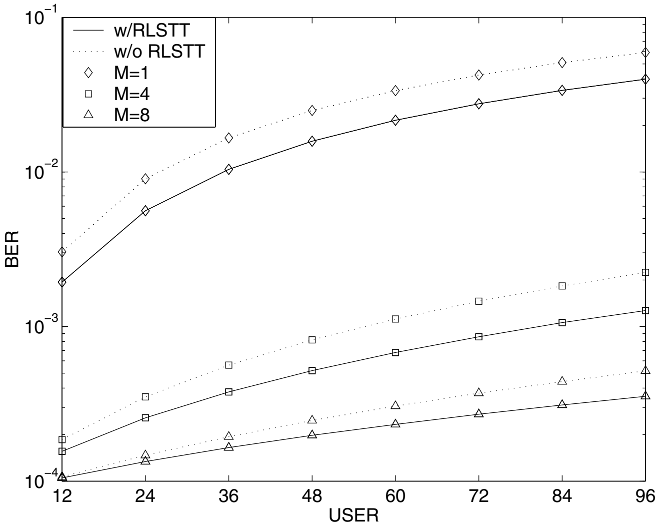


Fig. 5. AA w/ RLSTT vs. AA w/o RLSTT(δ=0.2, Eb/No=20[dB], $L_r = L^{(k)} = 2$)

5 Conclusions

We have demonstrated AA for DS-CDMA together with RLSTT scheme. While beamforming enhances the desired user signal and suppresses the cochannel interference from other directions, the RLSTT suppresses those whose arrival angle is the same at the desired user. Thus, the system capacity is enhanced and the performance for the demodulated data is improved. The results have shown that the RLSTT has superior performance and/or reduce the complexity of the system. We can use fewer numbers of antennas to achieve the performance of more antennas without RLSTT.

References

1. Lal C. Godara: Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Considerations. Proc. IEEE. **85** (1997) 1195-1245
2. A.F.Naguib: Adaptive Antennas for CDMA wireless networks. Ph.D dissertation, Stanford Univ., Stanford, CA. (1996)
3. G.Raleigh, S.N.Diggavi, A.F.Naguib, and A. Paularj: Characterization of Fast Fading Vector Channels for Multi-Antenna Communication Systems. Proc. 27th Asilomar Conference on Signals System and Computers. **II** (1994)
4. E.K.Hong, S.H.Hwang, K.J.Kim and K.C.Whang: Synchronous transmission technique for the reverse link in DS-CDMA terrestrial mobile system. IEEE Trans. on Communications. **47** (1999) 1632-1635
5. T.Eng and L.B.Milstein: Coherent DS-CDMA performance in nakagami multipath fading. IEEE Trans. on Communications. **43** (1995) 1134-1143

6. J.Litva and T.K. Lo: Digital Beamforming in Wireless Communication. Artech House Pub. (1996)

Appendix:

Evaluation of $\boldsymbol{E[\left\{C_{lh}^{(k,m)}\right\}^2]}$. From (8), We obtain the optimal Beamformer weight is given by

$$\mathbf{W}_l^{(k)} = \xi \cdot \left\{\mathbf{R}_{uu,l}^{(k)}\right\}^{-1} \mathbf{V}\left(\theta_l^{(k)}\right)$$

Since ξ does not affect the SINR, we can set $\xi = 1$. If the total undesired signal vector can be modeled as a spatially white Gaussian random vector, then $\mathbf{R}_{uu,l}^{(k)} = \left\{\sigma_{s,l}^{(k)}\right\}^2 \cdot \mathbf{I}$. Therefore, we can express the spatial correlation as

$$C_{lh}^{(k,m)} = \frac{\mathbf{V}^H\left(\theta_l^{(k)}\right) \cdot \mathbf{V}\left(\theta_h^{(m)}\right)}{\left\{\sigma_{s,l}^{(k)}\right\}^2} = \frac{CR_{lh}^{(k,m)}}{\left\{\sigma_{s,l}^{(k)}\right\}^2}$$

where $CR_{lh}^{(k,m)} = \sum_{i=0}^{M-1} e^{j\pi si\cos\left(\theta_l^{(k)}\right)} e^{-j\pi si\cos\left(\theta_h^{(m)}\right)}, \quad s = 2d/\lambda$
The second order characterization of the spatial correlation is given by

$$E[\left\{C_{lh}^{(k,m)}\right\}^2] = \frac{E[\left\{CR_{lh}^{(k,m)}\right\}^2]}{\left\{\sigma_{s,l}^{(k)}\right\}^4}$$

where

$$\begin{aligned}\left\{CR_{lh}^{(k,m)}\right\}^2 &= A\left(\theta_l^{(k)}, \theta_h^{(m)}\right)\\ &= \sum_{i=0}^{M-1} (i+1)\, e^{j\pi si\cos\theta_l^{(k)}} e^{-j\pi si\cos\theta_h^{(m)}}\\ &\quad + \sum_{i=M}^{2(M-1)} (2M-i-1)\, e^{j\pi si\cos\theta_l^{(k)}} e^{-j\pi si\cos\theta_h^{(m)}}\end{aligned}$$

The mean angle of arrival $\theta_l^{(k)}$ and $\theta_h^{(m)}$ have the uniform distribution in $[0,\pi)$ independently. So,

$$E[\left\{CR_{lh}^{(k,m)}\right\}^2] = \int_0^\pi \int_0^\pi A\left(\theta_l^{(k)}, \theta_h^{(m)}\right) d\theta_l^{(k)} d\theta_h^{(m)}$$

$$= \begin{cases} \sum_{i=0}^{M-1} (i+1)\, J_0(\pi si)\, J_0(-\pi si) + \sum_{i=M}^{2(M-1)} (2M-i-1)\, J_0(\pi si)\, J_0(-\pi si) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad ,k \;/\!\!=m \quad or \quad l \;/\!\!=h \\ M^2 \;\; ,k=m \quad and \quad l=h \end{cases}$$

Where $J_0(x)$ is the zero order Bessel function of the first kind.

Evaluation of $\left\{\boldsymbol{\sigma}_{s,l}^{(k)}\right\}^{2}$. From (7), the total interference plus noise for the l-th path of the k-th user in the matched filter output is given by

$$\mathbf{u}_l^{(k)} = \mathbf{I}_{l,si}^{(k)} + \mathbf{I}_{l,mai}^{(k)} + \mathbf{I}_{l,ni}^{(k)}$$

where

$$\begin{aligned}\mathbf{I}_{l,mai}^{(k)} = \sqrt{P/2} \sum_{\substack{i=1 \\ i \neq k}}^{K} \sum_{j=0}^{L^{(i)}-1} \beta_j^{(i)} \mathbf{V}\left(\theta_j^{(i)}\right) &\left\{ b_{-1}^{(i)}\left(j\right) RW_{ik}\left[\tau_{lj}^{(i)}\right] \right. \\ &\left. + b_0^{(i)}\left(j\right) \widehat{RW}_{ik}\left[\tau_{lj}^{(i)}\right] \right\} \cos\left[\psi_{lj}^{(i)}\right]\end{aligned}$$

$$\begin{aligned}\mathbf{I}_{l,si}^{(k)} = \sqrt{P/2} \sum_{\substack{j=0 \\ j \neq l}}^{L^{(k)}-1} \beta_j^{(k)} \mathbf{V}\left(\theta_j^{(k)}\right) &\left\{ b_{-1}^{(k)} RW_{kk}\left[\tau_{lj}^{(k)}\right] \right. \\ &\left. + b_0^{(k)} \widehat{RW}_{kk}\left[\tau_{lj}^{(k)}\right] \right\} \cos\left[\psi_{lj}^{(k)}\right]\end{aligned}$$

$$\mathbf{I}_{l,ni}^{(k)} = \int_{\tau_l^{(k)}}^{\tau_l^{(k)}+T} \mathbf{n}(t) g^{(k)}(t-\tau_l^{(k)}) a(t-\tau_l^{(k)}) \cos[\omega_c t + \psi_l^{(k)}] dt$$

If the undesired signal can be modeled as a spatially white Gaussian noise vector, then the covariance matrix $\mathbf{R}_{uu,l}^{(k)} = \left\{\sigma_{s,l}^{(k)}\right\}^2 \mathbf{I}$. In [4], we can obtain the total interference plus noise power as follow

$$\begin{aligned}\left\{\sigma_{s,0}^{(k)}\right\}^2 &= E_b T \varOmega_0 \left(\frac{(2N-3)(K-1)\{q(L_r,\delta)-1\}}{12N(N-1)} \right. \\ &\qquad \left. + \frac{\{q(L_r,\delta)-1\}}{4N} + \frac{\eta_0}{4E_b\varOmega_0} \right) \\ \left\{\sigma_{s,l}^{(k)}\right\}^2 &= E_b T \varOmega_0 \left(\frac{(N-1)(K-1)\,q(L_r,\delta)}{6N^2} \right. \\ &\qquad \left. + \frac{\{q(L_r,\delta)-1\}}{4N} + \frac{\eta_0}{4E_b\varOmega_0} \right) \quad , \quad for \quad l \geq 1\end{aligned}$$

A Noble Routing Algorithm for the Internet Computational GRID

Youngyong Kim

Dept. of Electrical and Electronics Engineering, Yonsei University
Shinchon-Dong, Seodaemoon-ku, Seoul 120-749, Korea
y2k@yonsei.ac.kr

Abstract. Computational Grid has become widespread and lead the way to high-performance computation using widely dispersed resources. The computational GRID architecture is very dynamic by nature with its distributed resources over many organizations, its heterogeneous resources, and pervasive availability with variety of tools. For the best performance of GRID computing, it is inevitable to find computational resources not only based on the computational availability but also based on the communication delay between server and client. In this paper, we suggest a noble routing protocol for the GRID computing based on communication delay. Our approach shows dramatic decrease on the overall computation time.

1 Introduction

Computational Grid has become one of dominant issues in recent distributed computing society. It is very demanding task to design optimal co-allocation of data, CPU and network resource for specific jobs. One of the major issues for the efficient design of computational GRID is to locate appropriate resource on highly heterogeneous and distributed environments. Although there has been a great deal of study on resource management in GRID architecture, less work has been done in relation to the communication bottleneck. In this paper, we propose noble routing architecture to locate adequate resources for the computation on the Internet scale Inter-GRID distributed environments.

Two major factors affecting the speed of GRID computing are communication latency and computational latency. Communication latency is defined as round-trip communication delay between the nodes which request a job and responds. This factor is especially important in the case that the given task is composed of small computational components whose time scale is comparable to the communication time scale. Computational latency is defined as the required computation time for the node to finish its assigned Job.

To minimize computation time, it is required to find node with the highest computational power and within the shortest communication distance. We consider the case of joint optimization for the resource location problem which finds node with the least cost where cost is defined as joint function of communication

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 33–37, 2002.

latency and computational latency for each requested job in very highly Inter-GRID environments. For efficient computational GRID, these two factors must be considered closely tied together. For instance, if one considers only computational resources, one might end up with super fast computer with very poor communication condition, (either permanently or temporarily) consequently leading to much longer overall computing time compared to the case when we use slower CPU nearby. We assume the general TCP/IP connectivity and modify the internet routing protocol to accommodate the computational GRID needs. We study the relationship of the considered problem with the constraint-based routing in the Internet. We also design message structure for the information which should be exchanged for nodes involving in GRID computing architecture and resource directory. A simulation for the proposed conceptual model shows the efficiency of computation with our proposed resource locating algorithm.

The remaining of this paper is composed as follow. Section 2 describes the proposed routing Algorithm. In Section 3, an analytical model to evaluate performance of proposed routing algorithm is established. In Section 4, our model is discussed in details. Conclusions are given in Section 5.

2 Routing Algorithm for the Computational GRID

2.1 Problem Formulation

One of the crucial parts is to find computational resource on the network, to perform some subtask. Let D_i be the deadline for the subtask i to be completed. Then our task is to find computational resource on the network which will finish the task within D_i . Recent Advances in fast processor design has brought down the cost of computation, has made it relatively cheap to get considerable computational power. However, communication link speed is not advancing as fast as computation power, and it is noticeable that communication bottleneck is by far more critical than the computational bottleneck. Therefore, it is imperative to find computational resource on the network not only based on the available computational resources but also based on the communicational delay.

In classical routing problem, the objective is to find shortest path from some source s to all destination, and each communication link is assigned a positive number called its length. Each path (i.e., a sequence of links) between two nodes has a length equal to the sum of the lengths of its links. A shortest path routing algorithm routes each packet along a minimum length (or shortest) path between the origin and destination nodes of the packet. In this paper, we consider routing protocol which considers shortest path not only based on communication link but also based on computational resource.

Suppose the deadline for sub-task i is given as D_i and expected workload as L_i then it is sufficient to find computational node j which has spare computational power C_j and meet the following condition,

$$D_i \geq 2 \sum_{k \in P_j} LD_k + L_i/C_j \tag{1}$$

where LD_k is the aggregated latency for the kth link on path P_j where P_j is the shortest path from source to the node j. The factor 2 is introduced because we need round trip delay for the server to get the computational result. Then our goal is to find node j which satisfies the following condition,

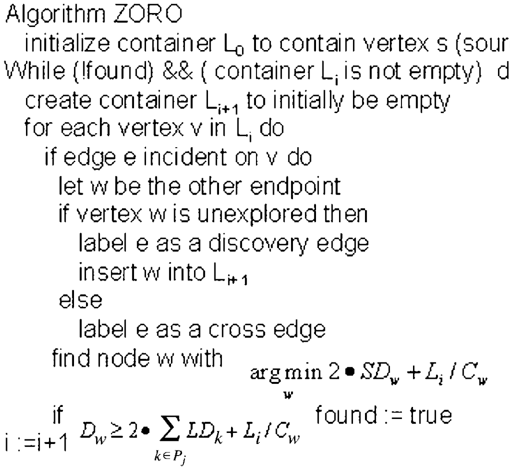
$$\mathbf{argmin_j} \sum_{k \in P_j} 2LD_k + L_i/C_j \tag{2}$$

for each sub-task i.

In next section, we develop the routing protocol to find the nodes which satisfies the condition as specified in (2).

2.2 Routing Algorithm

To find the node satisfying the condition (2) is not trivial task since the link latency is dependent on load condition on link i and computational capacity C_j is dependent on workload on node j. Therefore, we need some architecture to disseminate the required information. Hence we reached in the following modification of LSP(Link State Packets) in popular routing protocol as in OSPF. Firstly, we replace LD_k with WLD_k which implies the worst case delay on link LD_k.



```
Algorithm ZORO
  initialize container $L_0$ to contain vertex s (source); i := 1
While (!found) && ( container $L_i$ is not empty)  do
  create container $L_{i+1}$ to initially be empty
  for each vertex v in $L_i$ do
    if edge e incident on v do
      let w be the other endpoint
      if vertex w is unexplored then
        label e as a discovery edge
        insert w into $L_{i+1}$
      else
        label e as a cross edge
    find node w with  $\arg\min_{w} 2 \bullet SD_w + L_i / C_w$
    if  $D_w \geq 2 \bullet \sum_{k \in P_j} LD_k + L_i / C_w$  found := true
i :=i+1
```

Fig. 1. ZORO Algorithm

The argument behind this replacement is that dynamic measures in decision of route generally results in oscillation of selected routes and consequently leads to the instability of network.

With this modification, we can handle the link latency as constant so that we can use the off-the-self OSPF like routing algorithm. However, the processing latency in node j makes it impossible to use OSPF like routing algorithm since its cost is associated with the node itself not the link

Therefore, our algorithm considered here is not to pursue the optimal node which specifies the conditions in (2), but the suboptimal solutions which satisfy the sufficient condition in (1). Hence we get the following algorithm, called ZORO (ZOne based ROuting for the GRID). In this scheme we stop at the first node which satisfies the condition (1) without ever searching the optimal nodes as specified in (2). Our search is propagated like concentric circle as in Breadth First Search. The algorithm first search nodes within 1 hop, and find the nodes which satisfies condition (2). If the found node satisfies condition (1), our search ends. Otherwise, our algorithm continues to do the search within 2 hops, 3 hops, and so on. The beauty lies in the fact we separate routing algorithm and the computational requirement. Therefore, we can use any available off-the-self algorithm for the network routing to find distance and we only need to add computational latency later. Fig. 2 shows the mechanism of ZORO algorithm.

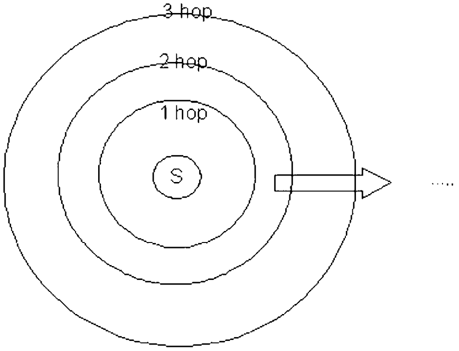


Fig. 2. How ZORO works

3 Performance Evaluation

We develop an efficient algorithm in section 2, and we discuss characteristics, performance, and practical requirement of our algorithm in this section.

3.1 Complexity Analysis

Our algorithm is based on general network routing. For instance, it is perfectly possible to use Dijkstra Algorithm and Distance Vector routing such as RIP. It is well known that the total running time of Dijstra algorithm is $O(|E|log|V|)$ where $|E|$ specifies number of edges and $|V|$ specifies the number of vertices. Since the shortest path is calculated only once throughout the algorithm, the complexity of our algorithm only adds the complexity of BFS(Breadth First Search) on which our algorithm is based on. For BFS, the complexity of traversal is $O(|E| + |V|)$ so that overall complexity can be considered as $O(|E|log|V| + |E|)$.

The exact analysis of mean path length to the selected is nontrivial job, since as we continue to fill out nearby nodes with workload, we will get further and further from the source and all resources are in very dynamic nature.

If we assume the link latency follows Exponential distribution with parameter λ and the computation latency follows Exponential distribution with parameter μ, from equation (2) the latency for the server to get response from node j will be sum of several exponential random variables resulting in Erlang distribution. Then its characteristic function is given by

$$F(s) = \prod_{i=0}^{k} \frac{2\lambda_k}{s + 2\lambda_k} \frac{\mu_i}{s + \mu_i} \tag{3}$$

3.2 Some Practical Consideration

For our algorithm to work it is imperative to distribute up-to-date load information for each load. However, it will be needed for any GRID architecture to work. Therefore, no additional overhead is needed from that. However, the difficult part is how to describe job(sub-task) correctly, i.e. to specify Li. From the practical view point, we may need some kind of semantics for the job descriptions.

4 Conclusion

In this paper we study the communication routing protocol within the framework of the GRID computational architecture. We propose a noble algorithm to find out computational node which jointly satisfies computational demands as well as within communication latency bound. We also give basic complexity analysis for our proposed algorithm. The future work includes the study of workload dynamics in GRID environments as well as practical design of routing packets.

References

1. S.J. Cox, M.J. Fairman, G. Xue, J.L. Wason, Keane A.J: The GRID: Computational and data resource sharing in engineering optimisation and design search. Proceedings of Parallel Processing Workshops. (2001) 207–212
2. D. Bertsekas, R. Gallager: Data Network. Prentice Hall (1992)
3. L. Kleinrock: Queueing Systems. Vol. 1. J. Wiley & sons. (1975)
4. M. Streenstrup: Routing in Communication Networks. Prentice Hall (1995)
5. M.F. Neuts: Matrix-Geometric Solutions in Stochastic Model. MD: Johns Hopkins Univ. Press, (1981)

A Genetic Routing Algorithm for a 2D–Meshed Fault–Tolerant Network System*

Sangmoon Lee[1], Daekeun Moon[2], Hagbae Kim[1], and Whie Chang[3]

[1] Department of Electrical and Electronic Engineering,
Yonsei University, Seoul, Korea
hbkim@yonsei.ac.kr
[2] Hyundai Heavy Industries Co., Ltd.,
Electro–Mechanical Research Institute, Korea
[3] C–EISA Co., Ltd., Seoul, Korea

Abstract. Unreliable data delivery due to network instability leads general network systems to serious situations. Especially in case of reliability–critical applications, a fault–tolerant routing is gaining more attention for guaranteeing complete data transmission even in the presence of faults. In the paper, a concise approach to the genetic design of the optimal fault–tolerant routing algorithm in a 2D–meshed network using GA(Genetic Algorithm) is proposed. The algorithm is based on the wormhole routing scheme with virtual channels for accomplishing deadlock–freedom. It permits the packets to return to the node where they previously stayed, and is adaptive to concurrently evolve its elementary strategies enhancing the associated performance. We perform computer simulations in a 10x10 meshed network to compare the proposed algorithm with the representative conventional one.

1 Introduction

In the distributed environment, the computers at each node perform certain programs with data that they shares or exchanges each other though the network. The network capacity is concerned with mainly the connecting and transmitting ability among nodes as well as the execution capacity of the computers. Especially, the connecting ability is fundamentally measured by both the message latency and the network throughput, which are intrinsically affected by the network topology and the routing strategy. The most important component that determines the nominal network capacity is the routing strategy. A certain fault(s) in the network causes the messages to deliver incompletely, moreover degrades its capacity. Therefore, it is necessary that the fault–tolerant routing scheme, which guarantees successful transmission of messages among non–faulty nodes in the presence of faulty components, takes responsibility for achieving high performance as well as reliability of a network system.

* This work is done by the MOCIE project; the Development of High-Performance Scalable Web Servers.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 39–46, 2002.

A few previous works on the fault–tolerant routing in the hypercubes[1,2,3,4] and the meshes[5,6,7,8,9] were reported. In [9], the authors extended the concept of virtual channels to a multiple and virtually–interconnected network, where a fault–tolerance scheme was equipped with increasing the number of virtual channels along the dimension by double, and extra channels were exclusively used to reroute packets around faulty nodes. One drawback is its infeasible requirement of too many virtual channels for implementation. In [6], a fault–tolerant routing algorithm was proposed by modifying the negative–first routing algorithm produced by the turn model in the mesh. However, the algorithm guarantees the aspects of fault–tolerance just against only one fault in the 2D–mesh. The authors of [8] used a partially–adaptive algorithm with adopting three virtual channels for acquiring fault–tolerance. If faults occur at edge nodes in the 2D–mesh, the algorithm regards all the nodes in a row containing the faulty node as being faulty. Its cost is substantial in H/W resources at the edge nodes of the mesh. In [5], a routing algorithm was also proposed to employ fault–tolerant features in a mesh network by means of both a fault–ring and a fault–chain consisting of fault–free nodes and links. However, the suggested algorithm is not fault–tolerant against arbitrarily–shaped faults.

In the paper, we propose the fault–tolerant routing algorithm in the 2D–meshed network for the standard topology on the basis of a wormhole routing scheme[10]. The developed scheme can be readily extended to other topologies with simple modification. The four virtual channels $\{vc_1, vc_2, vc_3, vc_4\}$ for accomplishing deadlock–freedom are also considered, where they are group(s) of channels sharing a physical channel while having their own queues[11]. We assume that each individual node recognizes only the faulty or non–faulty condition of its adjacent components (links and/or nodes). In the consideration of those, we apply GA(Genetic Algorithm) to developing the optimal fault–tolerant routing, and enhance certain network performance using GA method evolved in the paper. To validate the effectiveness of the proposed algorithm, which realizes the features of fault–tolerance as well as deadlock–freedom, it is compared with a conventional fault–tolerant routing algorithm, the fault–tolerant f–cube4[5].

2 A Fault–Tolerant Routing Algorithm Using GA

Since the wormhole routing scheme is inherently sensitive to the phenomenon of deadlock, it should be equipped with a certain scheme for deadlock–freedom. To overcome deadlock, we employ an approach of virtual channels, specifically four ones $\{vc_1, vc_2, vc_3, vc_4\}$. The usage of each virtual channel is determined by the locations of both source and destination nodes. And, the determined virtual channel is fixed until a message arrives at its destination. Table 1 illustrates the condition under which each virtual channel is utilized.

It is true that the presence and arrangement of faults seriously affect the routing strategy, whose effects are diminished by modifying the deactivation rules presented in [12]. Specifically, we convert fault regions into rectangular regions by the following rules: (i)If a node has three or more faulty nodes around it, it is

Table 1. Usage of virtual channels. Source node: $S(x_1, y_1)$, Destination node: $D(x_2, y_2)$. ('The location relationship of two nodes' AND($\wedge$) 'The utilized virtual channel')

$AND(\wedge)$	$y_1 < y_2$	$y_1 \geq y_2$
$x_1 \leq x_2$	vc_1	vc_2
$x_1 > x_2$	vc_3	vc_4

regarded as a faulty node and thus operates like the faulty node. We define this node by an unsafe node. (ii) If a node has two faulty nodes around it and if two faulty nodes around it are neighbors with each other, it is set to be an unsafe node as well. (iii) Including unsafe nodes to faulty nodes, both (i) and (ii) are iterated until every node in the mesh has at least one faulty node around it.

2.1 Routing Functions

We suppose that all of the nodes in the 2D–meshed network have the same routing table which basically determines the output direction of an arrived message at each node. To build the routing table encapsulating the message dispatching strategy, we consider not only the directions of both the previous and the destination nodes of a message but also the locations of faults. The node receiving a message has the information where to come from and where to go out for the message, and determines the direction to send it out by applying this information to the routing table. The directions of the previous node at any node are classified into four states; *Up, Down, Right, Left*, and the directions of the destination node into eight states; *Up–Left* (UL), *Up–Right* (UR), *Right–Up* (RU), *Right–Down* (RD), *Down–Right* (DR), *Down–Left* (DL), *Left–Down* (LD), *Left–Up* (LU). These classifications enable one to represent the routing function determining the output direction by:

$$D_{out} = R(D_{dest}, D_{pre}) \tag{1}$$

where D_{out}, D_{dest}, and D_{pre} are the output direction, the direction of the destination node, and the direction of the previous node, respectively. For instance, destination node 1 of Fig. 1 is *Up–Right* from the center node and destination node 2 is *Right–Down*, whereas the direction of the previous node from the center node remains fixed. Again, we can send the message with the destination of node 1 to the output direction that is read from (UR, *Left*) part of the routing table, and the message with the destination of node 2 to the output direction that is read from (RD, *Left*) part.

Since the whole mesh structure is not homogeneous, the number of physical channels connected with any node varies according to the location of the node. That is, the capacity that a node processes messages depends on its location. In particular, the edge nodes need the boundary managements as follows. If a read direction from the routing table points outside the mesh, we set the opposite direction to the read direction, and send a message to the set direction. However,

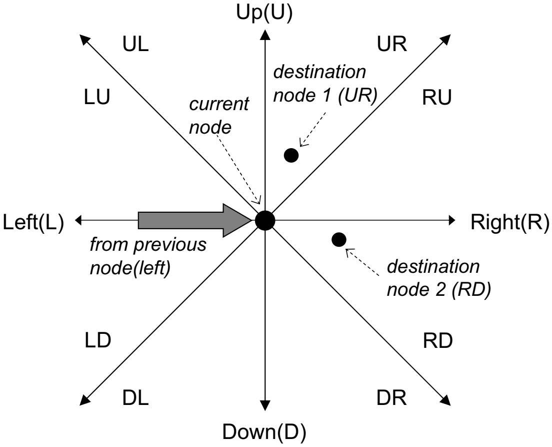


Fig. 1. Destination direction of a message transmitted from *Left*

if the node of the set direction is faulty, a message is sent to the opposite direction to the node at which a message previously stayed.

2.2 Parameter Representation: Crossover, Mutation, and Evaluation

The chromosome that capsulizes the parameters of the routing table is formed in a binary string. A routing table is divided into two subtables according to the fault arrangements of all nodes around the current node. One subtable represents when all nodes around the current node are non–faulty and its size is $5 \times 8 = 40$ cells, where the first number is relevant to the direction of the former residence node and the second number is associated with the direction of a destination node. The other subtable describes when only one of all nodes around the current node gets faulty and its size is $4 \times 8 = 32$ cells. The total size of the chromosome is, thus, $(40 \times 2) + (32 \times 2) = 144$ bits.

The table does not include all information about faulty states of the network, instead, it just describes the special case according to the number of faulty nodes around the current node. It is, however, quite reasonable because all the cases can be represented by transformation of one special case. The table needs adjusting through transformation when it is applied to a variety of faulty states of the network. In the paper, we conduct transformation through the rotation of the table according to the directions of faulty nodes, as illustrated in Fig. 2.

The primary operator of GA is crossover. The crossover operation consists of three steps. First, two chromosomes are selected from the parent pool. Secondly, the position where the crossover occurs along the two chromosomes is randomly chosen. Thirdly, all subsequent characters of two chromosomes after the crossover point are exchanged. A' and B' are two new strings according to this operation. String A' is made up of the first part of string A and the tail of string B. Likewise,

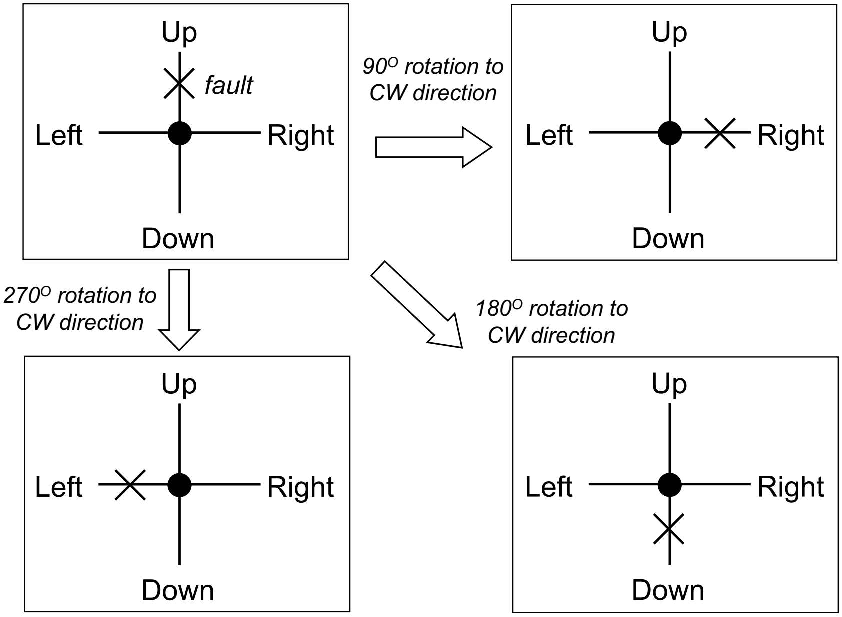

Up
fault
Left
Right
Down
90O rotation to CW direction
Up
Left
Right
Down
270O rotation to CW direction
180O rotation to CW direction
Up
Left
Right
Down
Up
Left
Right
Down

Fig. 2. Transformation through rotation of the table

string B' is composed of the first part of string B and the tail of string A. These newly–assembled chromosomes A' and B' are copied into the child pool.

The other important operator of GA is mutation. This enhances the capability of GA of searching for the near–optimal solutions[13]. The mutation is an occasional alternation of a value at a particular string position. It is an insurance policy against permanent loss of any simple bit. If a binary string A of length 8 is presented for mutation and the mutation point is selected at the second bit, the mutation operation converts the second bit 0 into 1, and the resultant A' is stored to the child pool. At the evaluation process, the performance of every parent is examined and stored as the fitness. The parent having a better fitness value enable the child to be generated with a high probability, which makes the fitness function (to properly evaluate the performance) quite important in searching for the optimal solution.

In the paper, every iteration generates the locations, where the message starts and ends, and the time, when this message randomly departs. At the same time, the information about fault locations (where the fault occurred) is generated as well. The performance of the routing table in the parent is, then, evaluated through the total latency of arriving messages, and stored as the fitness of the table into the parent. If the routing table is inadequately defined and thus the message can not arrive at the destination node, its performance is gradually decreased by the timeout, in which the table has a low probability to generate the child accordingly.

3 Computer Simulation

To properly validate the effectiveness of our suggested algorithm, we carry out time–step simulations at the flit level in a 10×10 meshed network, which are governed by the following specific conditions:

- Each message consists of 20 flits.
- The destination of each message is uniformly distributed.
- All nodes have the same probability of fault occurrences, and faults may occur anywhere.
- The network–clock is defined as the transmission time of a message from A to B, where either A or B is the neighbor node of each other.
- All messages are assumed to have the same priority.
- A message arriving at each node is immediately moved to a router via a certain processor.
- Each simulation according to applied loads and faults is performed for 20,000 network–locks enough to be statistically valid.

In fact, although the variations of these conditions produce different results to be trivial, those are expected to draw similar conclusions on the evaluation of our algorithm. The performance of the routing algorithm is measured in terms of two primary quantities; the average message latency and the normalized network throughput. The former is the elapsed time during which a message is transmitted from the source node to the destination node, whereas the latter is the number of messages delivered per network–clock over the number of messages that can be transmitted at the maximum load. Specifically, we consider the cases of 3% and 5% of the total network nodes being faulty in order to compare the performance of the fault–tolerant routing algorithm obtained by GA (f–GA) with that of the fault–tolerant f–cube4[5].

Fig. 3 and 4 show the network throughput and the message latency of f–cube4 and f–GA for both cases of 3% and 5% faulty nodes, respectively. In Fig. 3, the network throughput of f–GA is measured as 0.39, while that of f–cube4 is 0.36 for the case of 3% fault. In case of 5% fault, the network throughput of f–GA converges to 0.34, while that of f–cube4 does to 0.31. These verify that f–GA has not only a higher network throughput but also a lower message latency than f–cube4 under the same condition of environments, as shown in Fig. 4. In other words, we can conclude that f–GA is by far more efficient than f–cube4 for the network in the presence of certain faulty nodes. It can be implemented as a good solution for such applications as should be reliably and effectively operated even in harsh environments.

4 Conclusion

In the paper, we proposed the routing algorithm for fault–tolerant network system using GA. Although the partial presence of faults in the network, the algorithm is a highly–dependable to guarantee normal operation and/or transmission. To evaluate the optimal routing table applied to the fault–tolerant routing

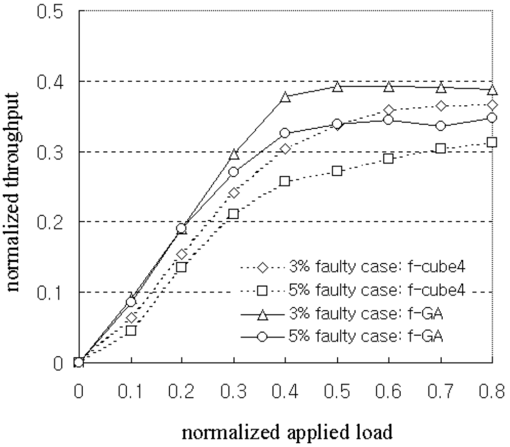


Fig. 3. Network throughput and the message latency. Normalized network throughput according to the normalized applied load in a 10×10 meshed network

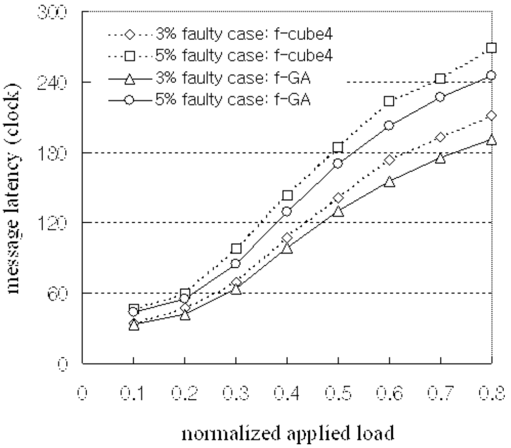


Fig. 4. Network throughput and the message latency. Average message latency according to the normalized load in a 10×10 meshed network

algorithm, followings are considered; (i) the routing table is presented by using chromosomes, and set the latency to a fitness function, (ii) the directions of both the previous and the destination nodes of a message are used to determine the output direction at any node, and (iii) all messages cannot return to the previous node. The ultimate routing table obtained according to these policies was verified to be effective enough to tolerate against component faults through simulation results. It is possible that the fault–tolerant routing algorithm using GA has a variety of extensibility if the detailed information about more complicated environments like the congestion control is appended to the chromosome.

References

1. ChIU G.M., etc: A fault–tolerant routing strategy in hypercube multicomputers. IEEE Trans. on Computers. **45** (1996) 143–155
2. KIM J., etc: Deadlock–free fault–tolerant routing in injured hypercubes. IEEE Trans. on Computers. **42** (1993) 1078–1088
3. LAN Y.: An adaptive fault–tolerant routing algorithm for hypercube multicomputers. IEEE Trans. on Parallel and Distributed Systems. **6** (1995) 1147–1152
4. SU C.C., etc: Adaptive fault–tolerant deadlock–free routing in meshes and hypercubes. IEEE Trans. on Computers. **45** (1996) 666–683
5. BOPPANA R.V., etc: Fault–tolerant wormhole routing algorithms for mesh networks. IEEE Trans. on Computers. **44** (1995) 848–864
6. GLASS C.J., etc: Fault–tolerant wormhole routing in meshes. FTCS–23. (1993) 240–249
7. DUATO J.: A new theory of deadlock–free adaptive routing in wormhole networks. IEEE Trans. on Parallel and Distributed Systems. **4** (1993) 1320–1331
8. CHIEN A.A., etc: Planar–adaptive routing: Low–cost adaptive for network multiprocessors. Proc. in 19th Ann. Int'l Symp. Computer Architecture. (1992) 268–277
9. LINDER D.H., etc: An adaptive and fault tolerant wormhole routing strategy for k–ary n–cubes. IEEE Trans. on Computers. **40** (1991) 2–12
10. DALLY W.J., etc: Deadlock–free message routing in multiprocessor interconnection networks. IEEE Trans. on Computers. **36** (1987) 547–553
11. DALLY W.J: Virtual–channel flow control. IEEE Trans. on Parallel and Distributed Systems. **3** (1992) 194–205
12. BOURA Y.M.: Fault–tolerant routing in mesh networks. Int'l Conf. on Parallel Processing. (1995)
13. CHUNG H.Y., etc: A self–learning and tuning fuzzy logic controller based on genetic algorithms and reinforcement. int'l Journal of Intelligent System. **12** (1997) 673–694

Allowable Propagation Delay for VoIP Calls of Acceptable Quality

Songun Na and Seungwha Yoo

Department of Computer Communication Engineering,
Ajou University, Suwon, Korea
swyoo@madang.ajou.ac.kr

Abstract. In VoIP network the primary factors affecting the voice quality are codecs, delay and any associated echo, and packet loss. In this paper, E-model is used to evaluate voice quality for end-to-end voice services over IP-based network. Assuming an acceptable voice quality objective is given on $R(\geq 70)$ scale, the minimum amount of propagation delay available to the connection is called as allowable propagation delay. The allowable propagation delay is much more important factor than the one-way delay because the packetization and jitter buffer delay and transport delay are almost constant for the codec in VoIP networks. Thus, the allowable propagation delay budgets for each codec are provided in order to offer voice calls of acceptable quality in the IP telephony system.

1 Introduction

Public Switched Telephone Networks (PSTNs) have evolved to provide an optimal service for time-sensitive voice applications that require low delay, low jitter, and constant but low bandwidth. That is, PSTN voice quality is relatively standard and predictable. IP networks were built to support non real-time applications such as file transfers or e-mail. These applications are characterized by their bursty traffic and sometimes high bandwidth demand, but are not sensitive to delay or delay variation. If PSTNs and IP networks are to converge, IP networks must be enhanced with mechanisms that ensure the quality of service (QoS) required to carry voice [1]. Because users of traditional telephone networks are used to quite high voice quality standards, providing comparable voice quality in IP networks will drive the initial acceptance and success of VoIP. Three elements such as clarity, end-to-end delay, and echo emerge as the primary factors affecting voice quality, particularly in the case of VoIP. The relationships among them can be quite complex as shown in Fig. 1. As the distance between the voice quality "point" and the intersection increases, voice quality decreases. However, no known mathematical relationship exists that can be used to derive a single voice quality number.

In the context of voice quality, clarity describes the perceptual fidelity, the clearness, and the non-distorted nature of a particular voice signal. In VoIP network, packet loss and voice codecs are major factors affecting the voice clarity.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 47–55, 2002.

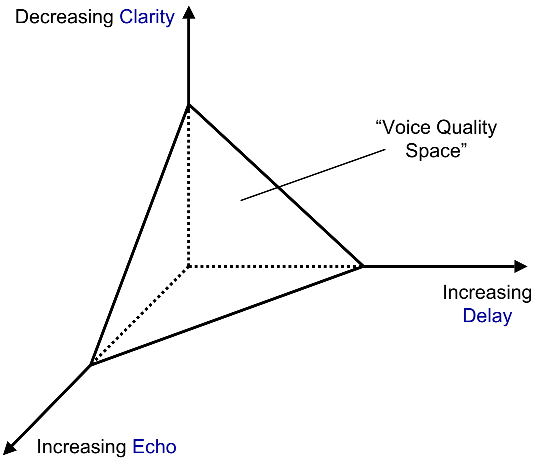


Fig. 1. Relationship among Clarity, Delay, and Echo with regards to Voice Quality

One of the challenges of transmitting real-time voice on packet networks is how to overcome the end-to-end delay encountered as the packet traverse the network. The delay is typically measured in milliseconds from the moment that the talker utters a word until the listener actually hears the word. This is termed as "mouth-to-ear" delay or the "one-way" delay which the users would realize in VoIP networks. In the traditional PSTN, the delay for domestic calls is virtually always under 150 ms [2]. At these levels, the delay is not noticeable to most people. Many international calls (especially calls carried via satellite) will have round-trip delay figures that can exceed 1 sec, which can be very annoying for users.

Though IP telephony is primarily used in a cost-reduction application, it should provide acceptable voice quality. What is considered acceptable quality of a VoIP call? As with most human factors, everyone has his or her own opinion on this issue. However, there is a definite limit of quality degradation that will be tolerated by users. The exact amount of quality degradation that will be tolerated is hard to define because users will balance the degradation of quality with the perceived value added by the system. Wireless telephone services are prime examples of where reduced connection quality will be accepted when balanced with the added value of high mobility.

Many factors influence the overall end-to-end voice quality of VoIP calls. If the factors are adequately controlled, it allows the more delay budget for a given voice quality level. Managing the delay budget is key to the success of VoIP services. A relationship between the delay and the other factors such as echo and packet loss should be studied to manage the delay budget. While ITU-T G.114 [2] made a recommendation on delay, this paper provides more flexibility than simple one-way delay limit like 150ms. In this paper, E-model [3-7] is used to evaluate speech transmission quality for end-to-end network connections in the VoIP network.

Section 2 introduces E-model and explains voice quality results with E-model. Section 3 describes the mouth-to-ear delay and the allowable propagation delay. Section 4 describes the sensitivity analysis of the allowable propagation delay. Section 5 concludes the results.

2 Computational Model

The E-model has been used as a computational tool to predict the subjective quality of a telephone call based on how it characterizes transmission parameters. The model was developed such that its results are in accordance with the results of extensive subjective laboratory tests. It combines the impairments caused by these transmission parameters into rating R, which ranges between 0 and 100. The relation between the different impairment values and R is given by

$$R = R_0 - I_s - I_d - I_e + A \; . \tag{1}$$

The term R_0 expresses the basic signal-to-noise ratio and the term I_s represents all impairments which occur more or less simultaneously with the voice signal, such as: too loud speech level, non-optimum sidetone, quantization noise, etc. The "delay impairment" factor I_d sums all impairments due to delay and echo effects, and the "equipment impairment" factor I_e represents impairments which are caused by low bit-rate codecs. The "advantage factor" A represents an "advantage of access" which certain systems may provide in comparison to conventional telephony systems. As an example, the advantage factor A for mobile telephony is 10 [3].

Fig. 2 shows E-model rating R to categories of speech transmission quality and to user satisfaction [6]. R below 50 indicates unacceptable quality. Among these factors, we have studied the impact of I_d and I_e on the quality of VoIP call, because R_0 , I_s , and A are not fundamentally different from the traditional PSTN calls. As far as one of the low-bit rate codecs is used, a VoIP call introduces more delay and distortion than a traditional PSTN call. First, the delay for VoIP calls is larger than that for traditional PSTN calls due to encoding, packetization, propagation, queuing, dejittering and decoding delay. Second, as a result of voice compression and packet loss during transport, the distortion of VoIP calls is not negligible. All connections below R=70 will suffer from some combination of distortion and long delay [6, 7]. The region between R=50 and R=70 encompasses the "Many users dissatisfied" and the "Nearly all users dissatisfied" categories in Fig. 2 and therefore deserves the low quality. In this paper, an acceptable quality category is then bounded by a lower limit of R=70. Fig. 2 illustrates the point by comparing the best-case curves for three popular IP codecs, G.711, G.729A [8] and G.723.1 (6.3 kbits/s)[9].

3 Allowable Propagation Delay

How much delay is too much? Delay does not affect speech quality directly, but instead affects the character of a conversation. Below 100ms, most users will not

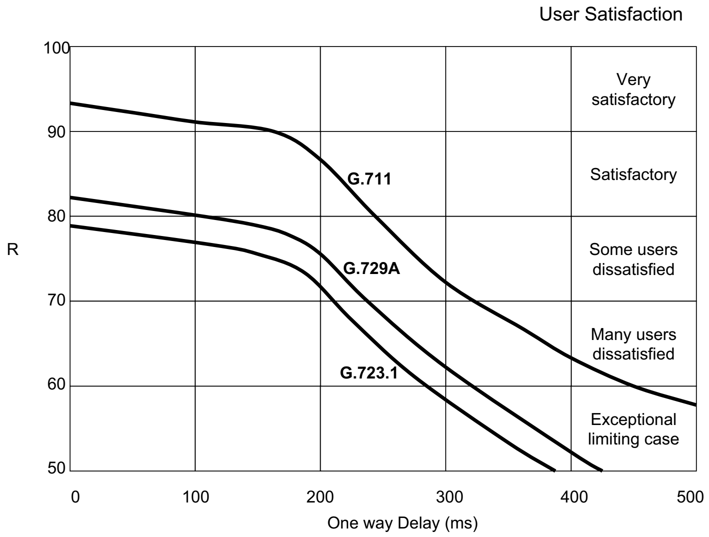


Fig. 2. Voice Compression Impairment

notice the delay. Between 100ms and 300ms, users will notice a slight hesitation in their partner's response. Beyond 300ms, the delay is obvious to the users and they start to back off to prevent interruptions.

3.1 Mouth-to-Ear Delay

In VoIP network, the mouth-to-ear (one-way) delay consists of three primary components; packetization and jitter buffer delay, transport delay, and propagation delay. If we let D_e, D_t, and D_p denote the packetization and jitter buffer delay , transport delay, and propagation delay, the one-way delay is given by

$$T = D_e + D_t + D_p = (D_{packet} + D_{jitter}) + D_t + D_p \ , \tag{2}$$

where D_{packet} is the packetization delay and D_{jitter} is the jitter buffer delay.

The packetization delay in a codec/vocoder is comprised of several components. The delay on the sender side includes the time taken to accumulate voice samples into a voice frame, the time required to compress the voice frame, the time to insert the voice frames into a packet and transfer the packet to the transport facility, and the firmware/hardware delays. In addition, some vocoders use a look-ahead function which waits for the first part of the following voice frame to provide information on how to help reconstruct the voice sample if there are any lost packets. The packetization delay on the receiver side consists of the time taken to decompress the voice frames into voice samples and the firmware/hardware delays. In addition, some codecs have an add-on packet loss concealment algorithm that adds some delay.

As the coded voice is being prepared for transport over the Internet, it needs to be assembled into packets. Looking inside a typical IP telephony data packet, each packet starts with an IP, UDP, and RTP header that totals 40 bytes. The decision of whether to pack more than one frame of data into a single packet is an important consideration for every IP telephony system. If a system is using the G.723.1, each packet would have 40 bytes of header and 24 bytes of data. That would make the header 167% of the voice data payload. If two frames are passed per packet, the overhead figure drops to 83%, with the side effect of adding yet another frame period of latency. A lost packet will result in speech clipping. In order to maintain a good end-to-end speech transmission performance, the voice contained in coded frames should be less than 64 ms of voice per IP packet [6]. Thus, the packetization delay D_{packet} is the time required for assembly of a packet including the encoding and decoding process, and is computed as

$$T_F(N_F+1)+T_L \leq D_{packet} \leq T_F(2N_F+1)+T_L \ , \tag{3}$$

where T_F is the frame size for the codec, N_F is the number of frames in a packet, and T_L is the look-ahead time for the codec. For G.723.1, the frame size is 30 ms and the look-ahead is 7.5 ms [9]. For G.729A, the frame size is 10 ms and the look-ahead is 5 ms [8].

Due to processing power limitations, IP telephone jitter buffers are typically frame-based meaning that the size of the buffer is a multiple of voice frame size. Frame-based jitter buffers can increase delay dramatically if the frame size is large. A rule of thumb for frame-based buffers is that the jitter buffer must be two times the voice frame size. Thus jitter buffer delay is computed as

$$D_{jitter} = 2T_F \times N_F . \tag{4}$$

3.2 Propagation Delay

Intranet carriers and corporate managed IP networks use equipment with only about 25 to 100 microsec of delay per hop, plus about 10 to 20 ms of jitter buffer delay end-to-end to accommodate source based jitter. Network based jitter is usually negligible relative to source based jitter. In this paper, the maximum transport delay is used as 20 ms. From equations (2), (3), and (4), the propagation delay can be bounded as

$$T-[T_F(2N_F+1)+T_L+2T_FN_F]-D_t \leq D_p \leq T-[T_F(N_F+1)+T_L+2T_FN_F]-D_t . \tag{5}$$

While the packetization delay D_e and the transport delay D_t are almost constant for the codec in the given IP network environment, the propagation delay D_p is variable in IP network. Assuming an acceptable voice quality objective is given on R scale, the minimum amount of propagation delay available to the connection is called as “allowable propagation” delay. The allowable propagation delay is much more important factor than the one-way delay because the allowable propagation delay is independent of codec and only dependent of VoIP network. Practically if the allowable propagation delay budget is less than 150 ms in IP network, it is not guaranteed to meet the acceptable voice quality objective.

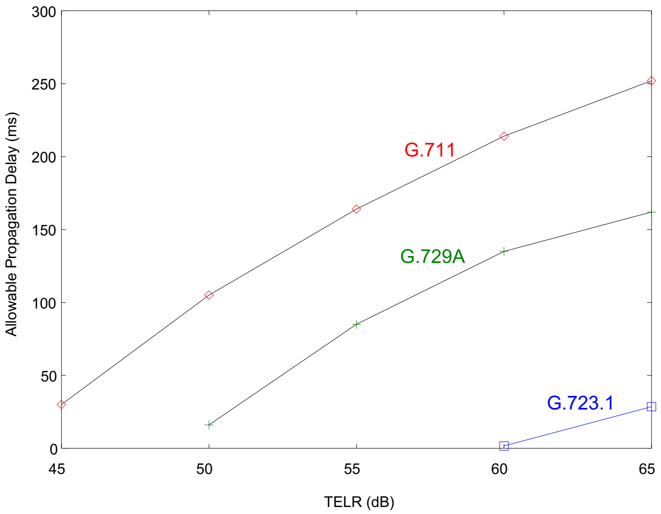


Fig. 3. Impact of Echo Cancellation on Allowable Propagation Delay

4 Sensitivity Analysis

The allowable propagation delay depends on the minimum voice quality level to which the connection is allowed to accept. It is assumed that the acceptable voice quality is $R \geq 70$ (the lower limit of the "Some users dissatisfied" category [6] in Fig. 2) in this paper. Managing the allowable propagation delay is key to meet the quality objective in the given environment. In this section the effect of the other factors such as echo cancellation, number of frames per packet and packet loss to the allowable propagation delay will be studied and three popular IP codecs (G.711, G.729A, and G.723.1) are considered.

4.1 Echo Cancellation

The family of curves in Fig. 3 shows the effect of echo cancellation as predicted by the E-model for the IP codecs. In Fig. 3, TELR is the sum of the echo losses around the loop, from one telephone set's transmitter back to the receiver on the same telephone set. It is shown that the allowable propagation delay is very sensitive to echo cancellation and drops as TELR decreases. This implies that as the echo cancellation is adequately controlled for G.711 or G.729A, the allowable propagation delay is good enough to meet the acceptable quality.

If there is no echo control, TELR is likely to be about 40dB for a traditional phone [10]. For G.711 with TELR=45dB, there is a very small propagation delay budget (30 ms) for the acceptable quality [7, 8]. Considering IP-based network, this delay budget is not good enough to provide the acceptable quality. However, if echo control is adequately controlled (for G.711 with TELR=65dB), there is a large enough propagation delay budget (252 ms) for the acceptable quality. It

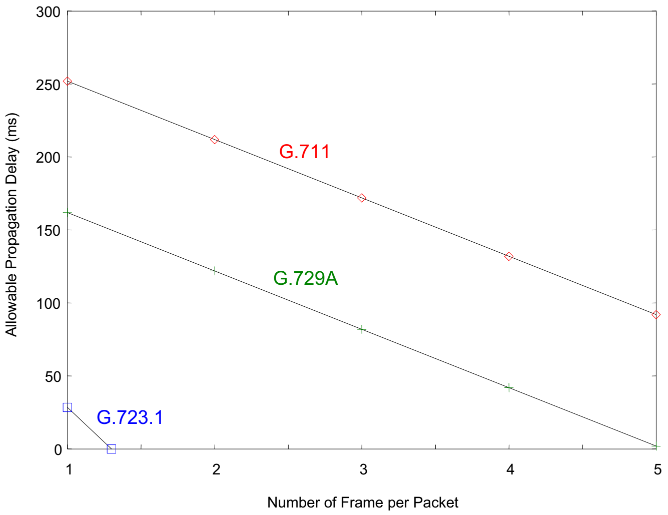


Fig. 4. Impact of Number of Voice Frames per Packet on Allowable Propagation Delay

turns out that echo cancellation at gateway and IP terminal should be provided for the acceptable quality. Especially, echo control is required for all types of IP terminals [11, 12].

For G.723.1, even if the echo cancellation is adequately controlled (TELR=65dB), there is a very small delay budget (28.5 ms) for the acceptable quality. Thus, it is not recommended to use G.723.1 because this budget is not good enough to meet the quality objective.

4.2 Number of Frames per Packet

The decision of whether to pack more than one voice frame into a single packet is an important consideration for IP telephony system. If two frames are passed per packet, the header overhead drops and instead the packetization delay increases. Also, a lost packet will result in speech clipping. In order to maintain the acceptable quality, it is recommended that the voice contained in coded frames should be less than 64 ms of voice per IP packet [11].

Fig. 4 shows that the allowable propagation delay is very sensitive to the number of voice frames per packet and drops rapidly as the number of frames increases. For G.723.1, one frame per packet is allowed to meet the acceptable voice quality. Hence, considering the allowable propagation delay budget, it is recommended that at most one frame of G.732.1, two frames of G.729A, or four frames of G.711, are assembled into a packet.

4.3 Packet Loss

For VoIP applications, it may be necessary to declare “very late” packets as lost to meet the acceptable delay limit. However, long one-way delay will increase the

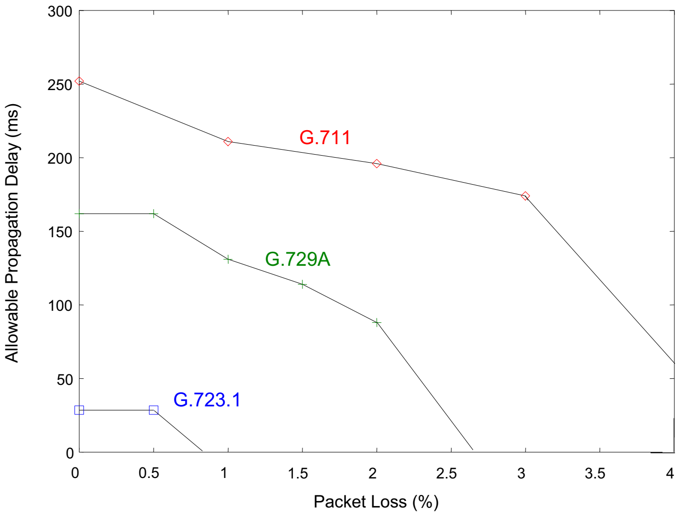


Fig. 5. Impact of Packet Loss on Allowable Propagation Delay

difficulty of conversation while packet loss will result in lower voice quality. The tradeoff between long delay and packet loss must be given careful consideration when designing VoIP networks.

Fig. 5 shows that the allowable propagation delay is very sensitive to packet loss ratio for G.711, G.729A, and G.723.1. As the packet loss rate is below 1%, the allowable propagation delay drops slowly. For G.729A, the propagation delay budget drops rapidly as the packet loss rate is above 2%. For G.711, the propagation delay budget drops rapidly as the packet loss rate is above 3%. Hence, considering the propagation delay budget, it is recommended that there are a packet loss budget of 1% for G.723.1, that of 2% for G.729A, and that of 3% for G.711.

5 Conclusion

In VoIP network the primary factors affecting the voice quality are end-to-end delay, packet loss, codecs and echo. The relationship among them can be quite complex and no known mathematical relationship exists. This paper provides more flexibility than simple one-way delay limit like 150 ms. If the primary factors are adequately controlled, it allows the more delay budget for a given voice quality level.

Assuming an acceptable voice quality objective is given on $R(\geq 70)$ scale, the minimum amount of propagation delay available to the connection is called as allowable propagation delay. The allowable propagation delay is much more important factor than the one-way delay because the packetization and jitter delay and transport delay are almost constant for the codec in VoIP networks. It is strongly recommended to use G.711 end-to-end among popular IP codecs because G.711 has a big allowable propagation delay budget. Also, it is not recommended to use G.723.1 because G.723.1 has a very small allowable propagation delay budget for the acceptable quality. It is shown that the voice quality of G.729A is much better than that of G.723.1 because delay and impairment of G.729A are smaller than those of G.723.1.

It is shown that the allowable propagation delay is very sensitive to echo cancellation and drops as TELR decreases. This implies that as the echo cancellation is adequately controlled, the allowable propagation delay increases. Also, considering the allowable propagation delay budget, it is recommended that at most one frame of G.732.1, two frames of G.729A, or four frames of G.711 are assembled into a packet.

Even if the echo and the number of frames per packet are adequately controlled, packet loss can be very harmful to the voice quality of VoIP calls and should be avoided in order to meet the acceptable quality if possible. Hence, considering the propagation delay budget, it is recommended that there are a packet loss budget of 1% for G.723.1, that of 2% for G.729A, and that of 3% for G.711.

Acknowledgement

This work was sponsored by MOCIE (Ministry of Commerce, Industry, and Energy) and BK21 program for the promotion of multimedia applications.

References

1. Saleem N. Bhatti and Jon C.: QoS-Sensitive Flows: Issues in IP Packet Handling, IEEE Internet Computing, (2000) 48–57
2. ITU-T Recommendation G.114: One-Way Transmission Time, (Feb. 1996)
3. ITU-T Recommendation G.107: The E-model, a Computational Model for Use in transmission Planning, (Sept 1998)
4. ITU-T Recommendation G.108: Transmission Systems and Media, Digital Systems and Networks, (Sept. 1999)
5. ITU-T Recommendation G.109: Definition of categories of Speech Transmission Quality, (Sept. 1998)
6. TIA/EIA/TSB116: Voice Quality Recommendation for IP Telephony, (Mar. 2001)
7. N.O. Johannesson: The ETSI Computation Model: A Tool for Transmission Planning of Telephone Networks, IEEE Communications Magazine, (Jan. 1997) 70–79
8. ITU-T Recommendation G.729: Coding of Speech at 8 kbits/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP), (Mar. 1996)
9. ITU-T Recommendation G.723.1: Dual Rate Speech Coder For Multimedia Communications Transmitting At 5.3 and 6.3 kbits/s, (Mar. 1996)
10. ITU-T Recommendation G.131: Control of Talker Echo, (Aug. 1996)
11. ITU-T Recommendation G.177: Transmission planning for voiceband services over hybrid Internet/PSTN, (1999)
12. D. De Vleeschauwer, J. Janssen, G.H. Petit: Delay Bounds for Low Bit Rate Voice Transport over IP Network, Proceedings of the SPIE Conference on Performance and Control of Network Systems III, Boston, USA, **3841** (Sept. 1999) 40–48

Robustness of a Neighbor Selection Markov Chain in Prefetching Tiled Web Data

Dongho Lee[1], Jungsup Kim[1], Sooduk Kim[1], Kichang Kim[1], and Jaehyun Park[2]

[1] Department of Computer Science and Engineering,
Inha University, Incheon 402-751, Korea
kchang@inha.ac.kr

[2] School of Information and Communication Engineering,
Inha University, Incheon 402-751, Korea
jhyun@inha.ac.kr

Abstract. The service speed of tiled-web data such as a map can be improved by prefetching future tiles while the current one is being displayed. Traditional prefetching techniques examine the transition probabilities among the tiles to predict the next tile to be requested. However, when the tile space is very huge, and a large portion of it is accessed with even distribution, it is very costly to monitor all those tiles. A technique that captures the regularity in the tile request pattern by using an NSMC (Neighbor Selection Markov Chain) has been suggested. The required regularity to use the technique is that the next tile to be requested is dependent on previous k movements (or requests) in the tile space. Maps show such regularity in a sense. Electronic books show a strong such regularity. The NSMC captures that regularity and predicts the client's next movement. However, Since the real-life movements are rarely strictly regular, we need to show that NSMC is robust enough such that with random movements occurred frequently, it still captures the regularity and predicts the future movement with a very high accuracy.

1 Introduction

Currently most web cache servers do not cache or prefetch map-like data because it is served through CGI mechanism and thus regarded as dynamic data. However, the use of map-like data is increasing, as in maps, large PDF files, electronic book, etc., and fast transmission of them is becoming important. A tile in map-like data is typically larger than a regular html file, and users usually request a number of tiles that are nearby in their locations, sequentially in one session. Retrieving these many tiles, each of them being relatively large, over the internet will cause significant delays in transmission time.

We want to predict the next move for a map-like data. We want a cache server to predict the next move for such data without the help from a web server or a map server even when the current page (or a tile in case of map-like data) is a first-timer. Previous techniques for prefetching cannot be applied directly to our problem, because they can deal with pages only if they have been requested repeatedly, or if they are first-timer pages for which web servers can give a help.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 57–65, 2002.

One promising solution is based on two observations. First, a tile has only a limited number of embedded links, and that among them only $2n$ links (for n-dimensional tile space) are important most of time. They correspond to the neighbor tiles of the current one. It is natural to expect that the next tile will be one of the neighbors of the current tile. For 1-dimensional tile space, there are two neighbors for each tile: left and right. For 2-dimensional space, there are four: left, right, up, and down. For n-dimensional space, there are $2n$ neighbors: two neighbors at each dimension. From this observation, we can form a candidate set for next move for any tile without parsing.

Secondly, a set of tiles belonging to the same map-like data show similar transition patterns among them, and we don't have to follow individual transition pattern for each tile. Instead, we can draw a single Markov chain, called Neighbor Selection Markov Chain (NSMC), for each map-like data that shows the general transition pattern of any tile belonging to it. Here, a state of this Markov chain does not represent a particular tile; instead, it represents k sequential movements that can be made, where k is determined by the characteristics of the particular map-like data. For example, a map-like data with 1-dimensional tile space may have a transition pattern such that the next tile is always the one in the right. For this data, it is appropriate to set k=1 because there is only one type of movement. In general, if the next movement depends on the previous k movements, the corresponding NSMC will have maximum $(2n)^k$ states. The cache server prepares these many states and builds edges based on the transition history[1]. With this Markov chain, the cache server can predict what will be the next movement from the current tile.

NSMC seems to show a good capturing capability for a regular transition pattern. However, the real life movement is rarely regular. It contains many unexpected movements, and we need to improve the NSMC so that it still can capture the real life movement. In this paper, we adjust NSMC and show it is robust enough such that under the influence of randomness it still captures the regularity of the movements.

The remainder of the paper is organized as follows. Section 2 surveys related works on prefetching and dynamic data caching. Section 3 discusses the proposed NSMC model. Section 4 summarizes the experiment and its results. Finally, Section 5 gives the conclusions.

2 Related Researches

Prefetching between a web server and a client was studied by pioneers in [1,2,3]. In [3], a web server monitors the pages it owns to calculate transition probabilities for the embedded links. When a client requests a page, the server knows which link will be requested next with what probability. The server pushes a list of possible candidate links with their transition probabilities to the client. The client, then, prefetches selected candidates from the web server while the current

[1] However, many of the states are never used, as shown in Section 4; thus the actual number of states is usually much smaller than $(2n)^k$.

page is being displayed. This technique requires both of the server and the client to be modified such that they behave correctly. Since we want to design a cache server that prefetches transparently without the cooperation from the web server or the client, this technique cannot be applied directly to our case. A Markov algorithm is used in [1] to determine related objects. The server monitors the reference pattern of each client separately and pushes related objects to the client appropriately. We also use a Markov algorithm but to extract a general reference pattern, not individual pattern.

Prefetching can be performed also between a cache server and a web server. The authors in [4,5] suggest a technique in which the web server collects usage statistics (the transition probabilities of embedded links) of the pages and relays the information to the cache server. As in [1,3], the web server has to be modified. A technique in [6] excludes the need to modify web servers by forcing the cache server to prefetch all embedded links within the currently requested page. To do this, however, the cache server has to parse each page to detect embedded links, and the overhead of fetching all of them is huge. Prefetching between a client and a cache server was discussed in [7]. The cache server collects the usage statistics for all pages it caches, and in order to do this, it parses each page remembering embedded links, and as requests are made, collects information about which link is referenced when and how often. The cache server pushes this information to the client, and the client decides which links will be prefetched. The client has to be modified to perform this prefetching.

Using Markov chains to predict the client's next request is studied in [8,9]. In [8], the expected number of accesses to each document within a specified time period from the current document is computed using a Markov chain. The highly scored documents will become possible candidates for prefetching. In [9], a Markov chain is built to store the transition probabilities among pages. Combined with the client's reference pattern, it can provide several value-added services like tour generation or personalized hubs/authorities generation. Both studies, however, try to compute exhaustive transition probabilities for all documents or pages, which is too expensive to apply to tile data.

The authors in [10] suggest a technique to speed up the retrieval time of geographical data where the query result is usually very huge in size. Traditionally, to cope with the huge size, the database application program executes an iterator-based algorithm as follows.

```
Result result = database.query(condition);
while (result.more()){
   doSomething(result.get());
}
```

The more() and get() are remote procedures to the database server, and calling them repeatedly incurs a high overhead. To reduce the overhead, they suggest using a cache server inside the client machine that will intercept the database query, open a persistent connection to the database server by itself, and supplies the query result to the client as if it is the database server. The client does not know about the intervening cache server and, through the same iterator-based interface, requests and receives the result. Since the cache server prefetches query results while the client is processing the current portion of the result, the retrieval time of whole results is improved. However, again, the client has to be modified, and the cache server in this case knows exactly what the client will request next: it does not have to predict.

3 NSMC and It's Robustness in Prefetching Tiles over Internet

3.1 Basic Idea

The references for tiles of a map-like data seem to show some spatial locality. When a tile has been requested, there is a high chance that one of the neighboring tiles will be requested next. The problem is which one, among the neighbors, will be requested next. The traditional prefetching techniques predict based on the reference history. They try to remember which neighbor was selected how many times from which tile. However, it is not easy to collect and maintain this information for all tiles. Furthermore, since we assume the number of tiles is very huge, there should be many tiles that are never referenced before. For these tiles, when they are referenced for the first time, how can we predict the next tile? In regular web data, the proportion of such first-timer data is about 30%[11]. We believe the proportion will be similar in tile data, and without handling properly these first-timer tiles, the prefetching would not achieve desired result.

Our observation is that if tiles belonging to the same map-like data show similar neighbor selection probabilities, we don't have to follow the reference history of each tile separately. Instead, we can try to capture the representative neighbor selection probabilities that can be applied to a general tile. Once we capture them, we can predict the next neighbor for any tile whether it has been referenced before or not.

3.2 Building a Neighbor Selection Markov Chain

In order to obtain representative neighbor selection probabilities, we build a Neighbor Selection Markov chain, NSMC. An NSMC has a number of states and connecting edges. A state of NSMC is a sequence of direction vectors that shows a tile selection history, while an edge is the probability to reach the target neighboring tile from the current one. For n-dimensional tile space, there are $2n$ neighbors for each tile because it has 2 neighbors, backward and forward, at each dimension. Diagonal neighbors can be expressed by the combination of these basic neighbors. Let's denote direction vectors for n-dimensional tile space as $d_1, d_2, \ldots, d_{2n}$. Direction vector d_i and d_{i+1} respectively represents the backward and forward movement to reach the two neighboring tiles at dimension $\frac{i+1}{2}$, for odd i. Also lets denote a tile at coordinates $(i_1, i_2, \ldots, i_n)$ as $t_{i1,i2,\ldots,in}$.

A tile selection history can be expressed as a sequence of direction vectors. For example, assuming 2-dimensional tile space, a tile selection history, t_{33}, t_{34}, t_{44},

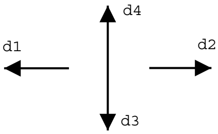


Fig. 1. 2-dimensional Direction Vectors

can be expressed as (d_4, d_2). The direction vectors at dimension 2 are shown in Fig. 1.

Now given an NSMC, predicting a next tile means deciding a next direction vector from the current state in an NSMC. The current state is defined as a sequence of direction vectors selected so far. To limit the size of NSMC, we restrict that the state can contain maximum k previous direction vectors, where k being a characteristic number specific to each map-like data. For n-dimensional tile space and direction vectors for each state, the number of possible states is $(2n)^k$. To build an NSMC, therefore, we need to build a Markov chain for these many states in maximum.

3.3 Hilbert Curve with Frequent Random Movements

Let's assume the client scans tiles along a Hilbert curve. An NSMC with n=2 and n=3 can be used to predict the client's movement. That is, without knowing about Hilbert curves, an NSMC can capture the regularity in client's movement. A Hilbert curve is being used to map a n-dimensional space on to a 1-dimensional space such that the adjacent objects in a n-dimensional space remain as adjacent as possible in a 1-dimensional space[12]. At level x, the number of tiles is 4^x, and the Hilbert curve at level x can be drawn by connecting four Hilbert curves at level (x-1) with appropriate rotation and change of sense of the two upper quadrants with no rotation and no change of sense, while the lower left quadrant being rotated clockwise 90 degree, and the upper right quadrant being rotated anti-clockwise 90 degree[12].

To show the robustness of NSMC under random movements, we incorporate randomness in the client's movement. We assume the client moves in Hilbert curve but with occasional randomness: the client makes a non-Hilbert move once in every 10 movements. To differentiate random movements from regular ones, we count the number of visits to each state. Those states with less than some threshold visiting counts will be considered as noisy states due to randomness and removed. We put the threshold at 10% of average visiting counts of all states. Then we build the NSMC with the remaining states. Fig. 2(b) shows the resulting NSMC for n=2 and k=3. Fig. 2(a) is the NSMC for the same n and k but with no random movements. Comparing with it, we can see the number of states grows from 28 in Fig. 2(a) to 33 in Fig. 2(b). With a random movement once in every 10 movements, the original number of states was 37. The average visit count per state was 258 for total 16384 Hilbert movements. The threshold

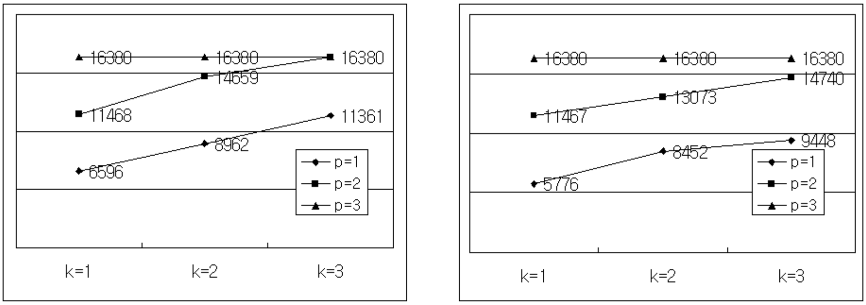
State	Path			Children States	
0	4	2	3	1	16
1	2	3	2	2	14
2	3	2	2	3	20
3	2	2	4	4	15
...	...	...	...	...	
25	4	1	3	10	
26	1	1	1	9	
27	3	3	3	17	

(a) n=2, k=3, regular movement

State	Path			Children States		
0	4	2	3	1	16	29
1	2	3	2	2	21	25
2	3	2	2	3	20	
3	2	2	4	4	31	15
...	...	...	...	...	...	
30	3	1	1	24	9	
31	2	4	4	27	13	
32	3	3	3	28	17	32

(b) n=2, k=3, random movement

Fig. 2. Neighbor Selection Markov Chain



(a) Regular movement

(b) Random movement

Fig. 3. NSMC Performance for 2-D Hilbert move

visit count, then, is 25.8. Three states had visit counts less than 25 and were removed. After removal, the number of states became 33 as in Fig. 2(b).

4 Experiments

To measure the NSMCs' prediction performance, we have generated a sequence of tile requests and feed them to the NSMCs. This time, however, we use the NSMC to predict the next request for each request and compare it to the actual one. The number of correct guessing is accumulated. We also allow the NSMC to select more than one candidate to see how the performance improves. The algorithm is given in Fig. 7. In the algorithm, p is number of candidates the NSMC can select. We have measured the performance of the NSMCs for $p = 1$, 2, and 3. The results are in Fig. 3 to Fig. 6.

Fig. 3(a) shows the performance of NSMCs for 2-dimensional Hilbert move. As k or p increases, we can see the prediction accuracy improves. For $k = 3$ and

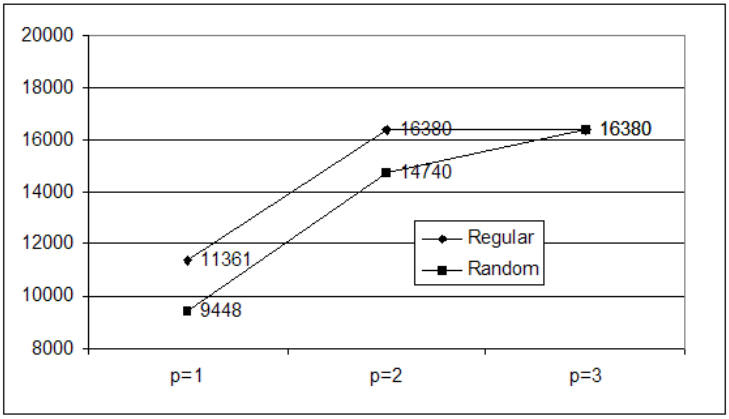
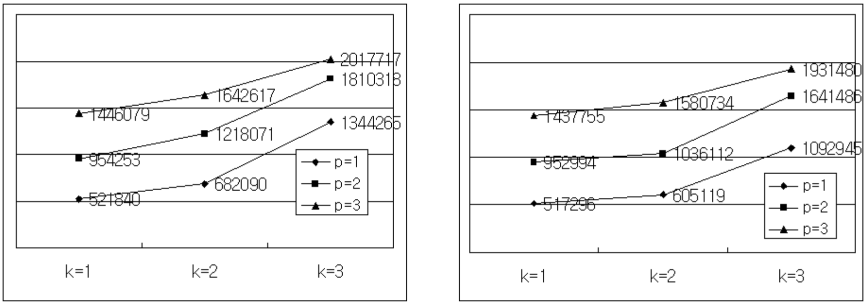


Fig. 4. Comparison of NSMC Performance between regular and random 2-D Hilbert movements



(a) Regular movement

(b) Random movement

Fig. 5. NSMC Performance for 3-D Hilbert move

p =2, the NSMC can predict perfectly; that is by monitoring three consecutive movements and by prefetching two candidate tiles, the NSMC always correctly guesses the client's next move. Fig. 3(b) shows what happens when the client moves randomly from time to time. With the randomness incorporated as explained in Section 3.3, the resulting NSMCs still show quite good performances. The prediction accuracy drops slightly, but with the same k=3 and p=2 it still shows 90% prediction accuracy. Fig. 4 compares regular and random NSMCs.

Fig. 5 to Fig. 6 show the performance of NSMCs for 3-dimensional Hilbert move with or without random movements. As the dimension increases, the number of possible neighboring tiles also increases making the correct prediction harder. However, the NSMC maintains rather high prediction accuracy for p=2 even for high dimensions.

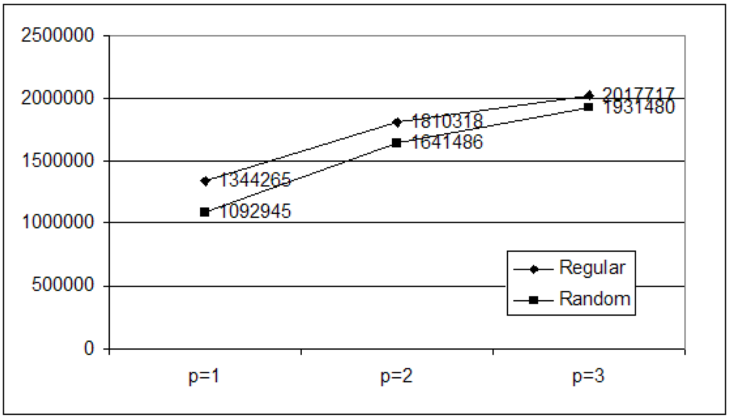


Fig. 6. Comparison of NSMC Performance between regular and random 3-D Hilbert movements

```
Input: hilbert_ordering[], k
Output: the prediction performance of NSMC for k
Method:
  curr_state = 0;
  hit = 0;
  for(i=k+1; i< side*side; i++){
     compute Mi;
     select  p children from curr_state;
     compute direction vector m1, m2,..., mp,
                                 to reach those children;
     if Mi is in (m1,m2,... mp) hit++;
  }
```

Fig. 7. Algorithm 1: Computing prediction performance of NSMC

5 Conclusions

In this paper, a technique that captures the regularity in the tile request pattern by using an NSMC (Neighbor Selection Markov Chain) even under real-life randomness has been suggested. When there exists regularity in the access pattern for internet data object, capturing and exploiting it for the purpose of prefetching is much cheaper than collecting and maintaining the reference histories of individual objects. Tiles for map-like data, e.g. maps, large PDF files, electronic books, etc., show such regularity in various degrees. We have shown that an NSMC can be used to capture such regularity that may exist in the tile request pattern. The required regularity is that the next tile can be predicted given the previous k tile requests. We have explained how to build an NSMC and measured its performance for a request pattern that follows a Hilbert curve. The experiments show that the NSMC predicts the future tile requests quite accurately even for high dimensional Hilbert movement and that it maintains its performance even under frequent random movements.

Acknowledgements

This work was supported by Inha University Research Grant (Grant number: 21374).

References

1. Bestavros, A., Cunha, C.: Server-initiated document dissemination for the www. IEEE Data Engineering Bulletin (1996)
2. Cao, P., Felten, E.W., Karlin, A.R., Li, K.: A study of integrated prefetching and caching strategies. In: Proceedings of 1995 ACM SIGMETRICS. (1995) 188–197
3. Padmanabhan, V.N., Mogul, J.C.: Using predictive prefetching to improve world wide web latency. In: Proceedings of ACM SIGCOMM Computer Communication Review. (1996)
4. Gwertzman, J., Seltzer, M.: The case for geographical push-caching. In: Proceedings of the Fifth Workshop on Hot Topics in Operating Systems. (1995)
5. Markatos, E.P., Chronaki, C.E.: A top-10 approach to prefetching on the web:technical report no. 173. Technical report, ICS-FORTH, Crete, Greece (1996)
6. Wcol Group: WWW collector – the prefetching proxy server for www. Technical report, ICS-FORTH, http://shika.aist-nara.ac.jp/products/wcol/wcol.html (1997)
7. Loon, T.S., Bharghavan, V.: Alleviating the latency and bandwidth problems in WWW browing. In: Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems. (1997)
8. Kraiss, A., Weikum, G.: Integrated document caching and prefetching in storage hierarchies based on markov-chain predictions. The VLDB Journal **7** (1998) 141–162
9. Sarukkai, R.R.: Link prediction and path analysis using markov chains. In: Proceedings of 9th International world wide web conference. (2000)
10. Chan, E.P., Ueda, K.: Efficient query result retrieval over the web. In: Proceedings of the 7th International Conference on Parallel and Distributed Systems. (2000)
11. Duska, B., Marwood, D., Feeley, M.: The measured access characteristics of world wide web client proxy caches. In: Proceedings of USENIX Symposium of Internet Technologies and Systems(USITS). (1997) 23–35
12. Jagadish, H.V.: Linear clustering of objects with multiple attributes. In: Proceedings of ACM SIGMODD 90 International Conference on Management of Data. (1990)

QoS Signaling for Parameterized Traffic in IEEE 802.11e Wireless LANs

Sai Shankar and Sunghyun Choi

Philips Research, USA
345 Scarborough Rd, Briarcliff Manor, NY 10510, USA
{sai.shankar,sunghyun.choi}@philips.com

Abstract. IEEE 802.11e Medium Access Control (MAC) is an emerging extension of the IEEE 802.11 Wireless Local Area Network (WLAN) standard to support Quality of Service (QoS). The IEEE 802.11e uses both centrally-controlled as well as contention-based channel access mechanisms to transfer data across the wireless medium. It also provides the mechanism to specify and negotiate the resource based on the user's QoS requirement. This paper presents a MAC-level QoS signaling for IEEE 802.11e WLAN and addresses its interaction with higher layer signaling protocols including Resource ReSerVation Protocol (RSVP) and Subnet Bandwidth Manager (SBM). We also explain a novel way of setting up sidestream connections for direct station-to-station streaming within an 802.11e WLAN.

1 Introduction

The world of tomorrow is full of multimedia and so efforts are gearing up to meet the global demand of multimedia, both real-time and non real-time, transfer across the integrated networks. The internet protocol (IP)-based Internet provides "best effort" data delivery by default without guaranteeing any service level to the users. A best effort service over the IP network allows the complexity to stay at the end-hosts, so that the network can remain simple. This scales well, as evidenced by the Internet to support its phenomenal growth. On the other hand, in recent years, IEEE 802.11 wireless LAN (WLAN) [4] has emerged as a prevailing technology for the (indoor) broadband wireless access for the mobile/portable devices. Today, IEEE 802.11 can be considered a wireless version of "Ethernet" by virtue of supporting a best-effort service. The IEEE 802.11 Working Group is currently defining a new supplement to the existing legacy 802.11 medium access control (MAC) sub-layer in order to support Quality of Service (QoS) [2,5]. The new 802.11e MAC will expand the 802.11 application domain by enabling such applications as voice and video services over WLANs.

The upcoming IEEE 802.11e will constitute the industry's first true universal wireless standard supporting QoS - one that offers seamless interoperability across home, enterprise, and public access networking environments, yet still offers features that meet the unique needs of each. Unlike other wireless initiatives, this is the first wireless standard that spans home and business environments by adding QoS features and multimedia support to the existing 802.11 standard,

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 67–83, 2002.

while maintaining full backward compatibility with the legacy standard. The QoS support for multimedia traffic is critical to wireless home networks where voice, audio, and video will be delivered across multiple networked home electronic devices and personal computers. Broadband service providers view QoS and multimedia-capable home networks as an essential ingredient to offering residential customers value-added services such as video on demand, audio on demand, voice over IP and high-speed Internet access.

In order to provide adequate service, some level of quantitative and qualitative determinism in the IP services is required. This requires adding some "smartness" to the network to distinguish traffic with strict timing requirements on delay, jitter and loss from others. This is what the protocols for QoS provisioning are designed to achieve. QoS provisioning does not create bandwidth, but manages it to be used more effectively to meet wide range of applications' requirements. The goal of QoS provisioning is to provide some level of predictability and control beyond the current IP "best-effort" service.

One very important component for the QoS support is the signaling protocol, which allows the end-hosts (and the intermediate nodes) of a given QoS session to communicate the desired QoS level and the corresponding resource amount. A number of end-to-end QoS signaling protocols in the IP layer and in the LAN environment have evolved to satisfy the wide range of application needs. The most well-known ones are ReSerVation Protocol (RSVP) [7] and its extension called Subnet Bandwidth Manager (SBM) [3] for the LAN environments. We outline these protocols and then describe how they fit into the IEEE 802.11e paradigm. The challenge of any QoS protocols is to provide differentiated delivery for individual flows or aggregates without breaking the network in process. Adding smartness to the network and improving the best-effort service represents a fundamental change to the network design that has made the Internet a great success.

The premise for this paper is based on the fact that there is a need for coordination between the 802.11e MAC and higher layers so that streaming applications can request and achieve their QoS requirements. Since the problems of the wireless domain are enormous, it is necessary to achieve some coordination between the MAC and higher layers to provide QoS. There needs to be a good transformation of the WLAN into a QoS network within end-to-end QoS context. In order to provide QoS, the roles and relationships between the higher layer protocols and IEEE 802.11e MAC needs to be clearly understood. The higher layer signaling protocols like RSVP and SBM perform macro management and the MAC performs micro management such as assigning different traffic streams to different queues and scheduling of service among different queues. In the above context, MAC layer signaling is very important to carry QoS information not only from higher layers to the MAC but also between different MAC entities. To avoid potential problems as QoS protocols are implemented in the network, the end-to-end principle is still the primary focus of all QoS architects. As a result, the fundamental principle of "leave complexity at the edges and keep the network core as simple as possible" is a central theme among QoS architectures.

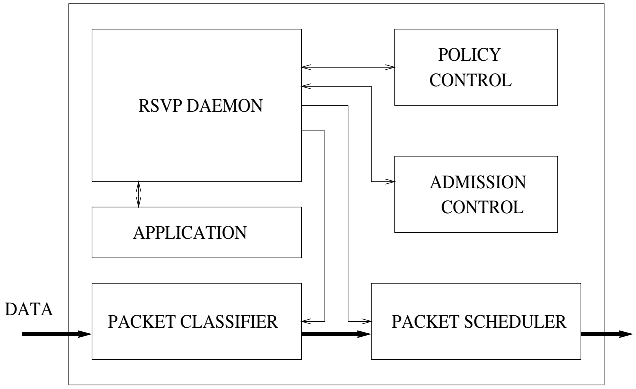


Fig. 1. RSVP daemon

Note that we address different types of signaling, i.e., the end-to-end signaling, MAC-level signaling for 802.11e, and internal signaling or interaction between the end-to-end signaling and MAC-level signaling within an 802.11e station. The actual QoS support during the run-time of a QoS stream, e.g., using a frame scheduling, is beyond the scope of the paper. The rest of this paper is organized as follows. Sections 2, 3, and 4 present an overview of RSVP, SBM, and IEEE 802.11e protocols, respectively. Section 5 gives the important terms and definitions that are taken from the present draft of IEEE 802.11e specification [2]. Section 6 explains the setup and deletion of QoS sessions accomplished through MAC signaling and finally, Section 7 concludes the paper.

2 Resource ReSerVation Protocol (RSVP)

ReSerVation Protocol (RSVP) [7], is a signaling protocol that provides reservation setup and control to enable the integrated service, which is intended to provide the closest model to circuit emulation on the IP networks. The RSVP is the most complex of all QoS technologies, for applications (hosts) and network elements (routers and switches). As a result, it also implements the biggest departure from the standard best-effort IP services and provides the highest level of QoS in terms of service guarantees, granularity of resource allocation and details of feedback to QoS enabled applications and users.

The host uses RSVP to request a specific QoS level from the network, on behalf of an application data stream. RSVP carries the request through the network, visiting each node that the network uses to carry the session. At each node, RSVP attempts to make a resource reservation for the session. The receiver specifies the QoS level with which it intends to receive the traffic stream from the source. Based on this information the intermediate nodes set aside the bandwidth required for that session. To make a resource reservation at a node, the RSVP daemon communicates with two local decision modules, i.e., admission control and policy control modules. The admission control module determines whether the node has sufficient available resources to supply the requested QoS. The policy control module determines whether the user has an administrative permission to make the reservation. If either check fails, the RSVP daemon returns an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a packet classifier and packet scheduler to achieve the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmissions to achieve the promised QoS for each session.

A primary feature of RSVP is its scalability. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. The reservation for a single receiver does not need to travel to the source of a multicast tree; rather it travels only until it reaches a reserved branch of the tree. While the RSVP protocol is designed specifically for multicast applications, it can also make unicast reservations. For details on the RSVP, the reader is referred to [7]. The process of the RSVP end-to-end signalling works as follows:

- Senders characterize the outgoing traffic in terms of the upper and lower bounds of bandwidth, delay and jitter via TSPEC (traffic specification). RSVP sends a PATH message with the TSPEC information to the unicast or multicast destination addresses. Each RSVP-enabled router along the downstream route establishes a PATH state that includes the previous source address of the PATH message.
- To make a resource reservation, receivers send a RESV (Reservation Request) message to the sender. In addition to the TSPEC, the RESV message includes a RSPEC (request specification) that indicates the type of service required, either controlled load or guaranteed, and a filter specification that characterizes the packets for which the reservation is being made such as transport protocol and port number. Together, the RSPEC and filter specification represent a *flow-descriptor* that routers use to identify each each flow or session. The RSPEC carries the the QoS values with which the receiver wants that connection. This is particularly applicable in a multicast environment wherein different receivers have different QoS requirements.
- When each RSVP router along the routing path from a receiver to the sender receives the RESV message, it uses the admission control process to authenticate the request and allocate the necessary resources. If the request cannot be satisfied because of lack of resources or authorization failure, the router returns an error back to the receiver. If accepted, the router sends the RESV message to the next upstream router.
- When the last router, i.e. the router between the source and the second downstream router, receives the RESV message and accepts the request, it sends a confirmation message back to the receiver. For the multicast case, it is the place where merging of flows occurs.
- There is an explicit tear-down process for releasing the reservation when sender or receiver ends an RSVP session.

2.1 Types of Service

RSVP enables two types of service, namely, the guaranteed and controlled load services:

- **Guaranteed service**: This comes as close to emulate a dedicated virtual circuit as possible. It provides firm (mathematically provable) bounds on end-to-end queuing delays by combining the parameters from various network elements along the routing path, in addition to ensuring bandwidth availability according to the TSPEC parameters.
- **Controlled Load:** This is equivalent to the best-effort service under unloaded conditions. Hence it is better than best effort, but cannot provide strict guarantees.

2.2 Characterization of Service

RSVP uses a token-bucket model to characterize its input/output queuing algorithm. A token-bucket is designed to smooth the flow of outgoing traffic, but unlike the leaky-bucket model, the token-bucket allows for higher data rates for short periods of time. The token-bucket parameters, token rate, bucket depth and peak rate are part of TSPEC and RSPEC (but the RSPEC parameters are different from TSPEC parameters. Based on both of the above parameters the router decides to set aside the bandwidth and other required resources). Here is a brief overview of the parameters:

- **Token rate** (r) : The sustainable rate for the flow measured in bytes per second. This reflects the average rate of the flow.
- **Token-bucket depth** (b) : The extent to which the data rate can exceed the sustainable average for short periods of time. The also indicates that the amount of the data sent over any time period t cannot exceed $rt + b$.
- **Peak rate** (p) : This represents the maximum sending rate of the source. More preciously, the amount of data sent over time period t cannot exceed pt.
- **Minimum policed size** (m) : The size of the smallest packet generated by the sending application. If the packet is smaller than m, it is treated to be of size m.
- **Maximum packet size** (M) : This is the size of the biggest packet measured in bytes.

As will be seen below, these parameters should be translated into the context of IEEE 802.11e QoS support.

3 Subnet Bandwidth Manager (SBM)

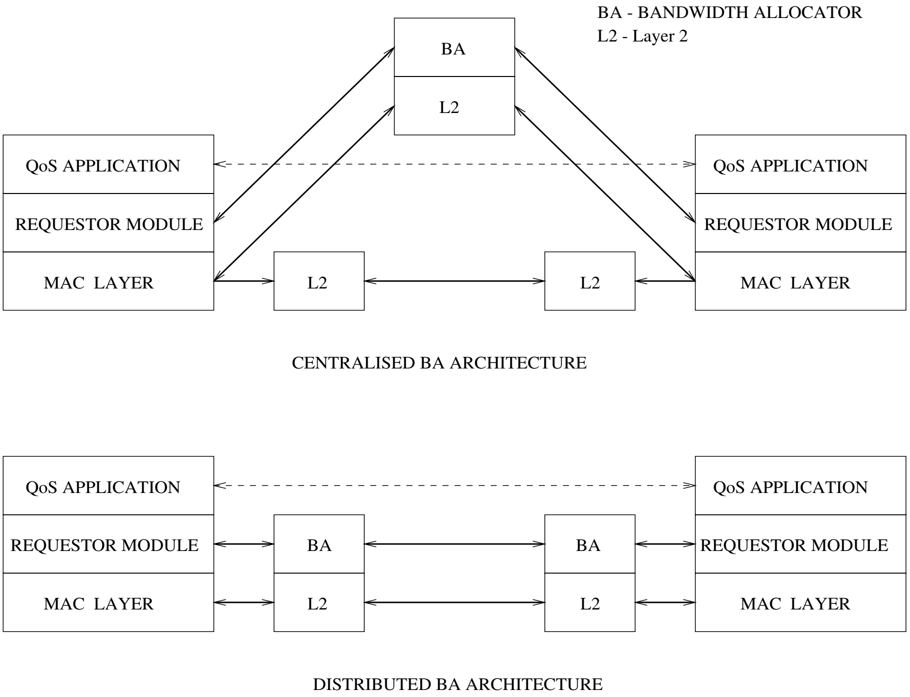
QoS assurances are only as good as their weakest link. The QoS session is end-to-end between the sender and the receiver, which means every router/bridge along the route must have support for the QoS provisioning. The sender and the receiver hosts must enable QoS so that the applications can enable it explicitly or the system can enable it implicitly on behalf of the applications. Each open systems interconnection (OSI) layer from the applications must be QoS-aware so that high priority traffic really receives high priority. The local area network (LAN) must enable QoS so that the high priority frames receive high priority treatment as they traverse the network media (e.g. host-to-host, host-to-router and router-to-router).

LANs (or a subnet) are normally composed of layer-2 and 1 networking devices such as Ethernet switches, bridges, and Ethernet hubs, and hence the whole such a LAN environment looks like a one hop to the layer-3 routers. As a shared broadcast medium or even in its switched form, layer-2 and 1 devices provide service analogous to the best-effort IP service in which variable delays can affect the real-time applications. However, IEEE has retro-fitted the layer-2 technologies to allow for QoS support by providing protocol mechanisms for traffic differentiation.

The IEEE 802.1D standards define how layer-2 bridge devices such as Ethernet switches can classify and prioritize frames in order to expedite delivery of real-time traffic [6]. The Internet engineering task force (IETF) for integrated services over specific link layers (ISSLL) has defined the mapping of upper layer QoS to layer 2 technologies. The mechanism for such a mapping is called Subnet Bandwidth Manager (SBM). SBM is a signaling protocol that allows communication and coordination among end-nodes, bridges, and routers (at the edges of the LAN) in a LAN environment by enabling the mapping of higher layer QoS protocols. The fundamental requirement in the SBM framework is that all traffic must pass through at least one SBM-enabled bridge. The primary components of the SBM are:

- **Bandwidth Allocator (BA)**: this maintains the states of the resource allocation on the subnet and performs the admission control according to the resources available.
- **Requester Module (RM)**: this resides in every end-host as well as in any bridges. The RM maps between layer-2 priority values and the higher layer QoS protocol parameters according to administrator defined policy. For example, if used with RSVP, it will map TSPEC, RSPEC or filter spec values to layer-2 priority values.

As illustrated in Fig. 2, the location of the BA determines the type of SBM architecture. There are two types of architectures, namely, centralized or distributed. Whether there is only one or more than one BA per network segment, only one SBM is called the designated SBM (DSBM). The DSBM may be statically configured or elected among the other SBMs. The SBM protocol provides an RM-to-BA or BA-to-BA signaling mechanism for initiating reservations, querying a BA about available resources and changing or deleting reservations. The SBM protocol is also used between the QoS-enabled application and the RM, but this involves the use of application programming interface (API) rather than



CENTRALISED BA ARCHITECTURE

DISTRIBUTED BA ARCHITECTURE

Fig. 2. SBM Implementation

the protocol. Therefore, it simply shares the functional primitives. A short description of the SBM protocol is outlined below.

- DSBM initializes and keeps track of the resource limits within its network segment .
- A DSBM client (i.e., any RSVP-capable end-host or router) looks for the DSBM on the segment attached to each interface. This is done by monitoring the ALLSBMAddress, which is the reserved multicast IP address 224.0.0.17.
- When sending a PATH message, the SBM client sends it to the DSBMLogicalAddress. This is a reserved multicast address given by 224.0.0.16 rather than to destination RSVP address.
- Upon receiving the PATH message, the DSBM establishes PATH state in the bridge, stores the layer-2 and layer 3 addresses from which it came, and puts its own layer-2/3 addresses in the PATH message. The DSBM then forwards the PATH message to next hop (which may be another DSBM or the next network segment).
- When sending the RSVP RESV message, a host sends it to the first hop, which is a DSBM taken from the PATH message.
- DSBM evaluates the request and if sufficient resources are available, forwards to the next hop or else returns a error message.

4 IEEE 802.11e MAC for QoS

Before addressing the signaling aspect of the 802.11e, we first briefly overview the core MAC operations in this section. An 802.11e WLAN, composed of an QoS access point (QAP) and one or more QoS stations (QSTAs), is called QoS Basic Service Set (QBSS). The 802.11e MAC defines a single coordination function called the Hybrid Coordination Function (HCF), which provides both a controlled and contention-based channel access mechanisms. The contention-based channel access of the HCF is often referred to as the enhanced distributed coordination function (EDCF) due to its root to the legacy DCF, i.e., the legacy 802.11 MAC [4]. The centralized coordinator called Hybrid Coordinator (HC) and is usually collocated in the QAP.

4.1 HCF Contention-Based Channel Access (EDCF)

The EDCF is based on a listen-before-talk protocol, called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), where a frame can be transmitted after listening to the channel for a random amount of time. It provides differentiated channel access to frames of different priorities as labelled by a higher layer. Due to the nature of the distributed contention-based channel access along with the uncertainty of the wireless medium, the EDCF cannot guarantee any rigid QoS. However, it provides so called "prioritized" QoS, which can be useful for applications that can live with statistical frame losses. With the EDCF, a single MAC can have multiple queues, which work independently, in parallel, for different priorities. Frames with different priorities are transmitted using different CSMA/CA contention parameters. That is, basically a frame with a higher priority is transmitted after listening to the channel for a probabilistically shorter period than frames with lower priorities. Note that there is no such a concept of stream supported by the EDCF. Each individual frame is treated relatively based on its corresponding priority.

4.2 HCF Controlled Channel Access

The controlled channel access of the HCF is based on a poll-and-response protocol in which a QSTA transmits its pending frames when it receives a polling frame from the HC. As the QSTAs contend for the channel according to the EDCF channel access, the HC is given the highest priority for the channel contention. That is, the HC is subject to winning the contention by listening to the channel for a shorter time than any other QSTAs before its transmission of a downlink frame or a polling frame. By polling a QSTA, the HC grants a polled transmission opportunity (TXOP) to the QSTA, where a TXOP represents a specific amount of time during which the polled QSTA, called the TXOP holder, assumes the control over the channel. The duration of a polled TXOP is specified in the particular polling frame. That is, during a polled TXOP, the TXOP holder can transmit multiple frames as long as the total duration for such transactions is not over the polled TXOP duration.

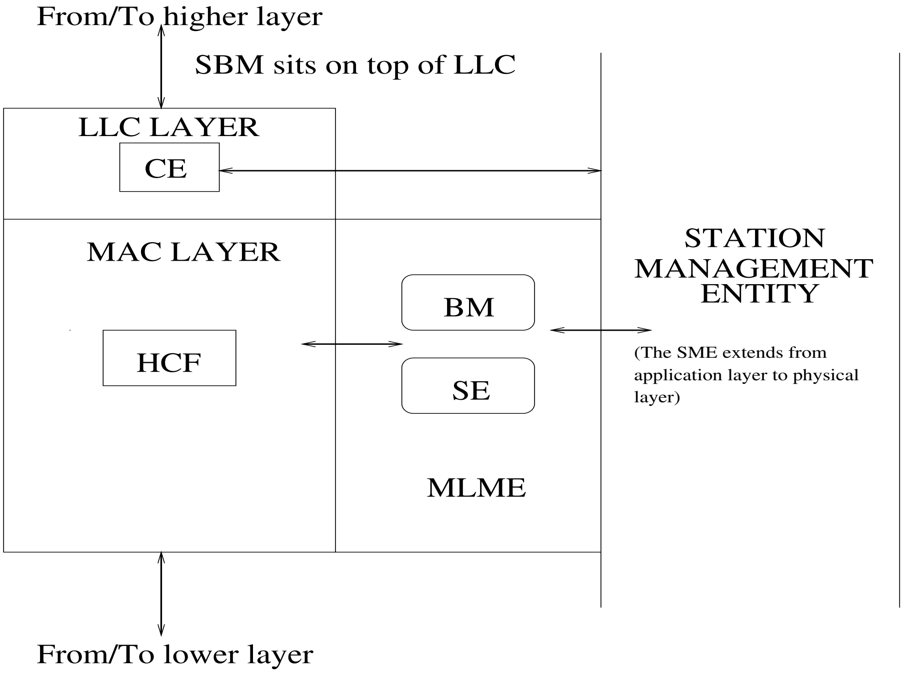


Fig. 3. MAC Architecture

Thanks to the centrally-controlled characteristics, the HCF can be used for the so-called "parameterized" QoS along with "prioritized" QoS. To support the parameterized QoS, the HC and the QSTA(s) set up a (layer-2 wireless link) stream along with the traffic characteristics and QoS requirements of the particular stream. Once such a stream is set up, the HC attempts to grant the TXOPs to the corresponding QSTAs (if the stream is from QSTA to QSTA or from QSTA to HC) or transmit the frames (if the stream is from HC to QSTA) according to the agreed specification. How to set up and maintain such a parameterized stream is handled by the MAC signaling as will be addressed in the following.

5 802.11e MAC Signaling

The 802.11e MAC defines two different types of signaling: one is the intra-STA signaling, and the other is Inter-STA signaling. One intra-STA signaling is defined between the station management entity (SME) and the MAC Layer Management Entity (MLME). The SME is a logical entity that communicates to all layers in the OSI stack while the MLME is a logical management entity for the MAC layer. Refer to Fig. 3 for the architectural overview of the relationship between the SME and the MLME. The inter-STA signaling is between two or more MAC entities within the same QBSS of 802.11e WLAN. For example, the communications between the HC and QSTAs using management frames for a stream setup belongs to this category. Another intra-STA signaling exists between the Logical Link Control (LLC) and the MAC.

5.1 Intra-STA Signaling

Signaling between LLC and MAC. Each data frame, that comes from the LLC to the MAC through the MAC Service Access Point (SAP), carries a priority value from 0 to 15. Within the MAC, this value is called Traffic Identifier (TID). The TID values from 0 to 7 specify the actual priority of the particular frame, in which the value 7 represents the highest priority and 0 the lowest priority. The frame with TID from 0 to 7 is served via prioritized QoS based on its priority value. The TID values from 8 to 15 specify a corresponding traffic stream which the particular frame belongs to. That is, such a TID is just a label of the corresponding stream, and the number itself does not tell anything related to the QoS level. Each frame belonging to a traffic stream is served subject to the QoS parameter values provided to the MAC in a particular traffic specification (TSPEC) agreed upon between the HC and the participating QSTA(s) of the stream.

Signaling between SME and MLME. The SME and MLME interact for a number of station/layer management activities such as starting a new QBSS, scanning the channel to find a new AP, and associating with a new AP. Out of all these different functions, we mainly address the interaction between the SME and the MLME for the QoS stream setup in this paper. The MLME of the QAP has two QoS-related entities, namely, the bandwidth manager (BM) and the scheduling entity (SE). The BM is responsible for keeping track of the wireless bandwidth and the scheduling entity is responsible for allocating TXOPs based on the requirements of different traffic streams.

The following MLME SAP primitives are newly defined for the signaling between the SME and the MLME as part of 802.11e to handle the traffic stream setup. Note that these MLME SAP primitives are used to support parameterized QoS as it requires a traffic stream setup. We present how these SAP primitives are used in setting-up stream connections in section 6.

- **MLME-ADDTS.request:** Sent by SME to MLME to initiate a stream management frame with specified parameters. This primitive requests addition or modification of a traffic stream with a specified peer MAC entity or entities capable of supporting parameterized QoS traffic transfer.
- **MLME-ADDTS.confirm:** Sent by MLME to SME to confirm the transmission of a stream management frame. This primitive informs the results of the traffic stream addition or modification attempt with a specified peer MAC entity or entities.
- **MLME-ADDTS.indication:** Sent by MLME to SME to inform the initiation of adding or modifying a traffic stream by another peer MAC entity. This primitive is signaled when a stream management frame is arrived from the peer MAC.

- **MLME-ADDTS.response:** Sent by SME to MLME to respond to the initiation of a traffic stream addition (or modification)by a specified QSTA MAC entity.
- **MLME-WMSTATUS.request:** Sent by SME to MLME to request the MLME for the amount of channel bandwidth available, channel status and the amount in use for QoS streams. This can be generated periodically or when a QoS flow is initiated or modified.
- **MLME-WMSTATUS.confirm:** Sent by MLME to SME to report the result in response to the MLME-WMSTATUS.request primitive.
- **MLME-SS-BW-QUERY.request:** Sent by SME to MLME to request the source QSTA to probe for the achievable transmission rate with the destination QSTA in the same QBSS. This primitive contains the frame size and the minimum physical layer transmission rate for the stream, both, derived from the RSVP PATH/RESV messages.
- **MLME-SS-BW-QUERY.response:** Sent by SME to MLME indicating the maximum transmission rate at which the source QSTA can sidestream to the destination QSTA in the same QBSS.
- **MLME-SS-BW-QUERY.indication:** Sent by MLME to SME to inform the initiation or result of probing for the achievable transmission rate for the sidestream connection by peer MAC entity. This primitive is signaled when a stream management frame is arrived from the peer MAC.

There are also MLME-DELTS.request, .confirm, .indication, and .response primitives defined to handle the tear-down process of a QoS stream. It should be noted that some primitives initiate a stream management frame while some others are signaled by receiving a QoS management frame. For example, MLME-ADDTS.request initiates a QoS stream management frame transmission while MLME-ADDTS.indication is generated when a QoS management frame is received. The actual transmission of the QoS management frame belongs to the external signaling as described below more in detail.

5.2 Inter-STA Signaling

Each single QoS data frame carries the TID value, which identifies the priority of the frame in case of the prioritized QoS or the corresponding traffic stream in case of the parameterized QoS. To carry such information, the 802.11e QoS data frame header is augmented by 2-octet QoS control field as shown in Fig. 4. The QoS control field uses four bits to indicate the TID value, and also carries some other QoS-related information. For example, the status of the queue, which the specific frame was dequeued from, is also indicated to aid the TXOP grant scheduling of the HC.

Two types of QoS management frames are defined for the inter-STA signaling to setup, modify, and delete traffic streams initiated by the corresponding MLME SAP primitives described in the previous subsection. The first type includes Add TS Request and Response QoS action frames used to set up or modify a QoS stream, and the second type includes Delete TS Request and Response QoS

OCTETS : 2	2	6	6	6	2	6	2	0 - 2312	4
Frame Duration	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	Frame Body	FCS

Fig. 4. Frame format of IEEE 802.11e QoS data

Element ID (13)	Length (30)	Source Address	Destination Address (6 Octets)	TAID (2 Octets)	TS Info (1 Octet)	Retry Interval (1 Octet)	Inactivity Interval (1 Octet)	Polling Interval (1 Octet)	Nominal MSDU Size (2 Octets)	Minimum Data Rate (2 Octets)	Mean Data Rate (2 Octets)	Maximum Burst Size (2 Octets)	TX Rate (1 Octet)	Reserved (1 Octet)	Delay Bound (1 Octet)	Jitter Bound (1 Octet)

Fig. 5. Traffic Specification Element

action frames used to delete a QoS stream. Each QoS action management frame includes the traffic specification (TSPEC) information element to communicate the corresponding QoS requirements and traffic specifications.

As shown in Fig. 5, the TSPEC element includes many quantitative objects of a traffic stream. Based on the values, the MAC layer attempts to reserve bandwidth for a particular stream and honor them if they are available. Many of the entities in this element are mapped directly from the higher layer needs, e.g., specified from the RSVP PATH/RESV messages after taking into consideration the MAC layer overhead and wireless channel conditions. Those include Nominal MSDU Size, Minimum Data Rate, Mean Data Rate, Maximum Burst Size, Delay Bound, and Jitter Bound. On the other hand, some entities such as TS Info, Retry Interval, Inactivity Interval, Polling Interval, and Tx Rate are more related to the different mechanisms of the MAC.

6 Interaction of RSVP/SBM and MAC Signaling

In this section, we present the interaction of RSVP, SBM, and the 802.11e MAC signaling for setting-up a parameterized connection. It is assumed that the QAP/HC hosts the DSBM [3]. We assume that SME and DSBM (or BA) within the HC/QAP can communicate while this detail is beyond the scope of this paper. Although SBM was originally designed to map incoming streams to 8 levels of priorities (similar to 802.11e prioritized QoS) as defined in IEEE

802.1D bridge specification [6], we use SBM to allocate bandwidth for parameterized QoS of the IEEE 802.11e WLAN. In case where the AP is connected to other 802 type networks, which can provide only the prioritized QoS based on 8 priority levels, the parameterized QoS is provided only in the 802.11e segment and not in other segments. This is not an unreasonable approach as the wireless segment is typically a bottleneck of the whole end-to-end network performance of a QoS session due to its relatively small and fluctuating bandwidth availability.

In this scenario, we consider a typical wired subnet wherein all the end-hosts are RSVP/SBM capable. Therefore, we use the signaling mechanism of the RSVP/SBM to route a QoS session in the wireless segment. Based on where the traffic originates and it is destined in the segment, three scenarios become important in the wireless environment. They are downstream signaling, upstream signaling and sidestream signaling. In the downstream signaling the source is a device that is connected to the wired environment and the destination is a QSTA in the QBSS. A stream is called upstream if the source is a QSTA and the destination is in the wired network. A stream is termed sidestream if the source and the destination are in the same QBSS and communicate to each other directly using the wireless medium.

We assume that all bandwidth reservations are done at the HC, which hosts the DSBM. This is very consistent in the sense that the HC has more knowledge than any other stations in managing the bandwidth in the wireless segment. In the following, we consider the connection setup cases only. Connection deletion is similar to connection setup and we use the signals MLME-DELTS.xxxx, where xxxx can be one of request, confirm and indication. This can be initiated by the receiver or source.

6.1 Downstream Signaling

Here we consider that the host in the wired network communicates to a QSTA of a QBSS via the HC/QAP of the QBSS. Therefore, the stream passes from the host in the Internet to the QSTA in consideration.

1. The RSVP at the wired host initiates a connection request for a QoS stream to be delivered to the QSTA through a PATH message. After travelling the wired network portion, the PATH message eventually reaches the DSBM, which is co-located with HC/QAP, and is in turn forwarded to the QSTA as a data type frame of 802.11e. The RSVP at the QSTA generates a RESV message in response to the PATH message and is transmitted to the DSBM at the HC/QAP.
2. The DSBM then requests the channel status update from the SME at the HC/QAP.
3. The SME, in the HC/QAP, in turn communicates to the MLME to obtain the information about the current channel status, which is kept track of by the BM residing in the MLME. The channel status is obtained using two MLME SAP primitives, i.e., MLME-WMSTATUS.request and MLME-WMSTATUS.confirm. The information on the channel status is passed to the SME, which in turn gives it to DSBM for making the admission decision.

4. The DSBM extracts the QoS parameters from new PATH/RESV messages for a downstream session, and makes the admission decision on the session (by accounting for channel status update from the MAC of HC/QAP via the SME).
5. If the session is admitted, the DSBM informs the SME that the session can be admitted and passes the source address (SA), destination address (DA) and TID values to the SME. The SME then establishes a stream identifier (SID) comprising of SA, DA and TSID Field for that session.
6. The SME also passes the SID and QoS values associated with the stream to the MLME for reserving resources via MLME-ADDTS.request. This information is used by the scheduling entity (SE) residing in MLME for scheduling TXOP during the run time for the admitted stream.
7. The MLME in turn sends an Add TS Request QoS action frame, containing the stream operation (Add) and QOS parameters. After sending the management frame, the MLME of HC/QAP generates a MLME-ADDTS.confirm to the SME.
8. Upon receipt of the management frame from the HC/QAP, the receiving QSTA checks the SID and QoS parameters of the new downstream. The MLME passes the above information to the SME through MLME-ADDTS.indication. If SME decides to accept the stream, it updates itself with the stream characteristics and initiates the MLME-ADDTS.response to HC/QAP. If the stream characteristics were not acceptable then the SME may initiate a delete operation as it is not able to accept the connection request.
9. Upon receipt of the positive response from the QSTA, the MLME at the HC/QAP passes the information to the SME through MLME-ADDTS.indication. The SME then informs the DSBM, which in turn forwards the RESV message to the source in LAN environment or to the next router.

6.2 Upstream Signaling

Here we consider that a QSTA is the initiator of the streaming connection and the recipient is in the wired Internet. The steam is going through the HC/QAP.

1. The RSVP at the QSTA initiates a stream connection by sending a PATH message. This PATH message is forwarded to the DSBM residing in the HC/QAP, which in turn forwards the PATH message to the next DSBM or router in the wired network.
2. If all the intermediate nodes have had enough resources to accommodate the requested connection, the DSBM will receive an RESV message from the wired network eventually. The DSBM on receipt of the RESV message contacts the SME of the HC/QAP for the current channel state information. It also extracts the QoS parameters for that stream from the PATH/RESV messages.

3. The SME of the HC/QAP obtains the channel state information from the MLME using two MLME SAP primitives, i.e., MLME-WMSTATUS.request and MLME-WMSTATUS.confirm. Upon receiving the channel state update from the MLME, the SME which in turn passes that information to the DSBM. Based on the information obtained from SME, the DSBM makes the admission decision.
4. If the DSBM decides to admit the session, it contacts the SME for confirmation and informs it that the session can be admitted and passes the source address (SA), destination address (DA) and TID values to the SME.
5. The SME of HC/QAP passes the SID (comprising the SA, DA, and TID) and QOS parameters to the MLME for bandwidth allocation via MLME-ADDTS.request. The MLME in turn generates a Add TS Request QoS action management frame for the upstream session. After that the MLME sends MLME-ADDTS.confirm to the SME.
6. Upon receipt of the Add TS Request QoS action management frame, the source QSTA passes the QoS parameters to SME through MLME-ADDTS.indication. If the SME decides to admit the stream, it updates itself with the stream parameters, and sends the Add TS Response QoS action frame by indicating it. If not, the negative response is sent back to the HC/QAP either for renegotiation or for dropping the connection request.
7. Upon receipt of the positive ADD TS Response QoS action frame, the MLME of the HC/QAP informs its SME through MLME-ADDTS.indication. The SME then informs the DSBM that the connection is accepted. The DSBM then forwards the RESV message to the source QSTA.

6.3 Sidestream Signaling

Here both the source and destination QSTAs are in the same QBSS. We propose that the HC/QAP determines whether the communication between the two QSTAs will be sidestream or relayed via the HC/QAP. This decision is important not only for the routing information but also for conserving bandwidth of the wireless medium. The channel state information has to be determined in a different way as the HC/QAP needs to know whether the two QSTAs can communicate to each other directly at the rate the sending QSTA wants to transmit. The advantage of sidestream is that it conserves bandwidth by transmitting traffic directly rather than relaying the same stream via the HC/QAP. In the latter case the bandwidth consumed is twice than sidestream transmission assuming that the same transmission rate is used in the physical layer for uplink and downlink.

1. The RSVP from the source QSTA initiates a PATH message. This PATH message is forwarded to the DSBM residing at HC/QAP instead of the destination QSTA.
2. The DSBM receives the PATH message and forwards the PATH message to the destination QSTA. The destination QSTA initiates the RESV message, which is forwarded to the DSBM.

3. The DSBM after receiving the RESV message will contact the SME of the HC/QAP for the channel state information. Since it is a communication between two stations in the same QBSS, the HC/QAP will try to determine if it is desirable for the source QSTA to sidestream to the destination QSTA as it may be more bandwidth-efficient. The decision whether it will allow the source QSTA to sidestream it or upstream it is left to HC/QAP.
4. The SME of HC/QAP will make its MAC generate an action frame to the source QSTA by asking it to initiate the channel status update. This is done through the MLME SAP primitive called MLME-SS-BW-QUERY.request. This frame has the nominal frame size and the minimum physical layer transmission rate information that is required for the stream.
5. To obtain the channel state information, the SME in the source QSTA initiates a maximum transmission rate probing. Based on the nominal frame size, it generates packets at the highest rate and expects the acknowledgment from the receiver. If the receiver responds, then that rate is assumed to be the achievable physical layer transmission rate between the QSTAs. If the acknowledgment is not received the channel status probe sequence is repeated by transmitting the frames at a lower rate up to the minimum transmission rate informed by the HC/QAP. QSTA performs the update to determine the rate and then relays that information to the HC/QAP through a response action frame. This is done through MLME-SS-BW-QUERY.response.
6. The response is passed from MLME to SME of HC/QAP through MLME-SS-BW-QUERY.indication. The SME at the HC/QAP on receipt of the information makes the decision whether to admit the request as sidestream or as upstream/downstream. If the minimum transmission rate is not achievable, the sidestream cannot be established, and accordingly upstream/downstream can be the only candidate. The decision is passed to the DSBM.
7. The DSBM makes then forwards the RESV message to to the sending QSTA for updating the RSVP connection.

Note that for the sidestream the TSPEC element to have the receiver address indicating whether the stream passes through the HC or directly to the destination QSTA.

7 Conclusion

In this paper, we present the IEEE 802.11e MAC signaling and explain how different signaling primitives can be fit together to work in integrated services (IntServ) over the IEEE 802.11e WLAN. We also present a novel way of establishing sidestream sessions in a wireless environment. Experimenting the above signaling mechanism is considered by the authors for the future work.

References

1. Kandala, S., Kowalski, J. M.: Signaling for streaming in IEEE 802.11e. IEEE 802.11-01/301, (May 2001)
2. IEEE 802.11 WG: Draft Supplement to Part II: Wireless Meduim Access Control (MAC) and Physical Layer (PHY) specifications, Medium Access Control (MAC) Enhancements for Quality of Service (QoS). IEEE 802.11e/D2.0a, (Nov. 2001)
3. Ghanwani, A., Pace, W., Srinivasan, V., Smith, A., Seamen, M.: A Framework for Integrated services over switched and shared 802 networks. IETF RFC 2816, (May 2000)
4. IEEE 802.11 WG: Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Reference number ISO/IEC 8802-11:1999(E), IEEE Std. 802.11, 1999 edition, (1999)
5. Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G., Stibor, L., IEEE 802.11e Wireless LAN for Quality of Service. Proc. European Wireless '2002, Florence, Italy, (Feb. 2002)
6. IEEE 802.1 WG: Part 3: Media Access Control (MAC) Bridges. Reference number ISO/IEC 15802-3: 1998, ANSI/IEEE Std 802.1D, 1998 Edition, (1998)
7. Braden, R., Zhang, L., Berson, S., Herzon, S., Jamin, S.: Resource ReSerVation Protocol (RSVP) - Version 1:Functional Specification, IETF RFC 2205, (Sept. 1997)

Content-Adaptive Request Distribution Strategy for Internet Server Cluster

Hagyoung Kim[1], Sungin Jung[1], and Cheolhoon Lee[2]

[1] Computer System Division,
Electronics and Telecommunications Research Institude, Taejon, Korea
{h0kim,sijung}@etri.re.kr
[2] Parallel Processing Laboratory,
Department of Computer Engineering, Chungnam National University,
Taejon, Korea
chlee@ce.cnu.ac.kr

Abstract. This paper addresses a distribution strategy for an Internet server cluster where the content-adaptive distribution is performed by each of the front-end node in the cluster. The system architecture considered here is a hybrid one consisting of a set of logical front-end dispatcher nodes and a set of back-end server nodes. Each front-end node in the cluster may service a request locally or forward it to another node based on the request content. This paper suggests a new distribution strategy called CARD (Content-Adaptive Request Distribution) that assigns most frequently used files to be hot which is served locally on each front-end node, while making the rest of the files to be partitioned and served among the back-end nodes. We present and evaluate the optimal configuration and hot size. The approach takes into account the file access patterns and the cluster parameters such as the number of nodes, node memory, TCP handoff overheads, data consistency overheads and disk access overheads. The simulation results show that the CARD achieves a linear speedup with the cluster size and that the CARD outperforms both the traditional centralized and distributed strategies, and outperforms a pure partitioning and replication strategy.

1 Introduction

Servers based on clusters of nodes are the most popular configuration used to meet the growing traffic demands imposed by the World Wide Web. A cluster of servers, arranged to act as a single unit, provides incremental scalability as it has the ability to grow gradually with demand. However, for clusters to be able to achieve the scalable performance, it is important to employ the mechanisms and policies for balanced request distribution.

In this paper, we first describe various methods for building scalable Internet server clusters. Also, we analyze the limits to scalability and the efficiency of these methods, and the trade-offs between the alternatives. Then, we propose a hybrid dispatcher architecture where the dispatcher components have the same

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 85–94, 2002.

routing information in all front-end nodes. In this approach, this routing information is updated on a content-adaptive workload information basis. We then propose a new request distribution strategy called CARD that takes request properties into account and identifies a small set of most frequent file to be hot. The requests to the hot files are processed locally by the front-end server in the cluster, while the other files are partitioned and served among the back-end servers. The goal of the CARD strategy is to minimize the forwarding overhead caused by TCP handoff for the most frequent files, while optimizing the overall cluster memory usage by partitioning the rest of the files that typically contribute to the largest portion of the working set. We also suggest an algorithm that identifies the optimal configuration and the size of hot for a given workload and system parameters. In this way, the CARD tunes itself to maximize the performance for a given access pattern and cluster configuration parameters.

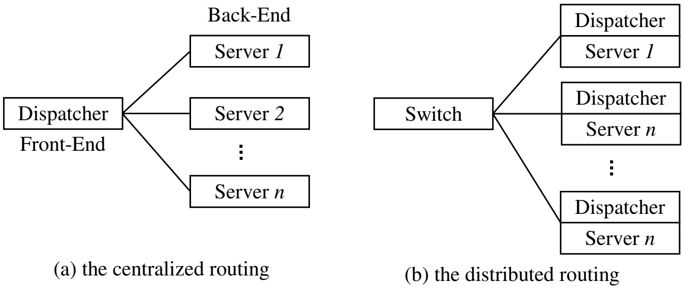
Simulation results show that the CARD achieves a linear speedup with the cluster size. It also shows excellent performance improvements compared to the traditional centralized/distributed strategies and a pure partitioning and replication strategy. The remainder of the paper presents our results in more detail. Section 2 formally introduces related works, Section 3 describes CARD strategy with the emphasis on the adaptive-analysis algorithm. The simulation results and their analysis are given in Section 4. Finally, concluding remarks appear in Section 5.

2 Related Works

Load balancing solutions can be represented by two major groups: 1) DNS based approaches, 2) redirection based approaches. Early works on distribution and assignment of incoming connections across the cluster server [9, 11] has relied on Round-Robin DNS(RR-DNS) to distribute incoming connections across the cluster of servers. This is done by providing a mapping from a single host name to multiple IP addresses. Due to DNS protocol intricacies, RR-DNS was found to be of limited value for the purposes of load balancing and fault tolerance of scalable Internet servers. The second group, redirection based approaches, employs a specialized front-end node, the load-balancer, which determines the least loaded server to which the packet is to be sent [3, 8].

Fig. 1(a) shows the most typical cluster configuration with a single front-end. Rather than delegating to DNS the responsibility of distributing requests to individual servers in the cluster, several research groups have suggested the use of a local router to perform this function. Fig. 1(b) shows a decentralized dispatcher architecture where the dispatcher on each server has the same routing information. In the centralized routing, the load distribution is done by a single node, the dispatcher. These approaches are less reliable than the decentralized ones. However, decentralized approaches have the problem of a heavy communication overhead incurred by frequent information exchange between nodes.

Content-adaptive request distribution, on the other hand, takes into account the content(URL, URL type, or cookies) when making a decision to which back-



(a) the centralized routing (b) the distributed routing

Fig. 1. Two Internet server cluster architectures

end server the request is to be routed. Previous works on the content based request distribution [2, 4, 7, 10] has shown that policies distributing the requests based on cache affinity lead to significant performance improvements compared to the strategies taking into account only load information. To explain the cluster architecture implementing the content-adaptive balancing strategies, we first define the terms, front-end, back-end, and dispatcher as follows. A front-end node is a server that runs the dispatcher and processes HTTP request, if necessary. A dispatcher runs on a front-end node and determines a server to process each client request, and forwards the request to it, if necessary. A back-end node is a server which processes the HTTP requests forwarded. To be able to distribute the requests based on the content, the dispatcher should implement some mechanisms such as TCP handoff [7] and TCP splicing [6]. In [2], the authors compared performance of both mechanisms showing the benefits of the TCP handoff schema.

In this paper, we assume that the dispatcher implements the TCP handoff mechanism. The switch in front of the cluster can be a simple LAN switch or L4 level load-balancer. For simplicity, we assume that the clients directly contact a dispatcher, for instance via RR-DNS. In this case, the typical client request is processed in the following way: 1) the client web browser uses the TCP/IP protocol to connect to a dispatcher. 2) the dispatcher accepts the connection and parses the request. 3) the dispatcher hands off the connection using the TCP handoff protocol to the server selected. 4) the selected server sends the response directly to the client.

The results in [2] show good scalability properties of the above architecture when distributing requests with the LARD(Locality-Aware Request Distribution) policy [7]. The main idea behind LARD is to logically partition the documents among the cluster nodes, with the objective of optimizing the usage of the overall cluster memory. Thus, the requests to the same document will be served by the same cluster node that will most likely have the file in memory. However, each node statistically will serve only $1/N$ of the incoming requests locally and forwards $(N-1)/N$ of the requests to the other nodes using the TCP handoff mechanism. The TCP handoff is an expansive operation leading a significant forwarding overhead, decreasing the potential performance benefits of the proposed solution.

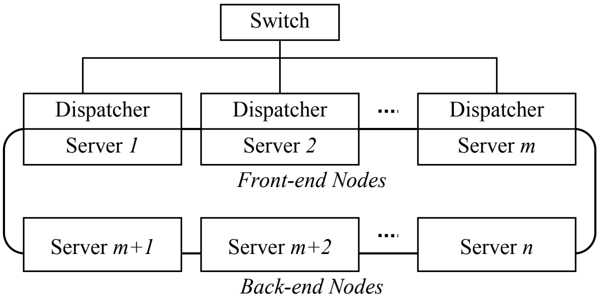


Fig. 2. Scalable Internet server architectures

3 Content-Adaptive Request Distribution Strategy

3.1 Architecture

In this section, we propose a new hybrid dispatcher architecture as shown Fig. 2, where a dispatcher reside on many, if not all, nodes in the cluster. The nodes of the cluster are divided into two logical categories: 1) front-end nodes and 2) back-end nodes.

The main component of a content-adaptive load balancer consists of dispatchers, and TCP handoffs. A dispatcher receives a clients HTTP requests and maps the URL to a real server. Each dispatcher uses several data structures including cluster table, URL table, and mapping table. The cluster table has each server's IP and MAC addresses, a timer for failure detection, and load information. The URL table stores the document location, size, format, etc by URL. The mapping table is the mapping information of real servers for serving client requests. After the dispatcher maps a specific client to a real server, the TCP handoff sends and receives a control message with a real server via the TCP handoff protocol. Using the control message, the TCP connection between the front-end and the client is taken over to a real server so that the real server can receive HTTP requests from the client without additional 3-way handshaking.

We use several system parameters in determining which server to take over the client requests. To minimize the response time for each request, we suggest a heuristic based on the estimated processing cost for each request using the following formula.

$$T_{response} = T_{forward} + T_{data} + T_{CPU} + T_{network} \tag{1}$$

$T_{forward}$ is the cost to redirect the request to another node, if required. T_{data} is the time to transfer the required data from the disk. T_{CPU} is the amount of processing time required to compute the task. $T_{network}$ is the cost for transferring the results over the Internet. To minimize $T_{response}$, we concentrate on minimizing $T_{forward}$ and T_{data} using the proposed hybrid architecture and Content-Adaptive Request Distribution Strategy. To reduce communication and memory overhead incurred by information exchange and content replication, we consider the optimal number of front-end nodes using the following formular.

$$m = (Workload_{hot}/Workload) \cdot n \tag{2}$$

Here, when m = 1, it becomes a centralized architecture and when m = n, a decentralized architecture. $Workload_{hot}$ is the workload of hot based on the estimated data, Workload is the workload of all requests, and n is the number of node in the cluster.

3.2 Content-Adaptive Request Distribution Strategy

It is well known that the performance of an Internet server greatly depends on efficient memory usage. The performance is much better when it reads pages from memory than disks. If all the files of the site fit in memory, the Internet server demonstrates an excellent performance because only the first request for each file will require a disk access, and all the following file accesses will be served directly from memory.

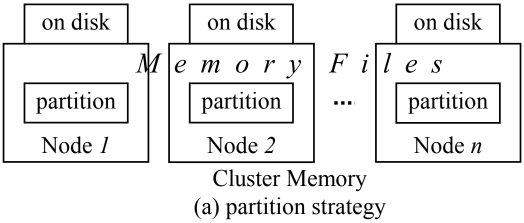
In applying the CARD strategy, we attempt to achieve the following two goals: 1) to minimize the forwarding by identifying the subset of hot files to be processed on the front-end nodes, i.e. allowing the replication of these files in the memories across the front-end nodes, 2) to maximize the number of requests served from the total cluster memory by partitioning files to be served by different back-end servers.

The key of the proposed CARD strategy is the use of information on the frequencies and sizes of individual file, called f and s, respectively. Let Frequency-Size be the table of all accessed files with their frequency and the files sizes. This table is sorted in decreasing frequency order. The algorithm presented in this section assumes that the cache replacement policy of the file cache has the property that the most frequently used files will most likely be in the cluster memory. Hereafter, we define ClusterMemory as the total size of all the file caches in the cluster.

If all the files were partitioned across the cluster, the most probable files to be in the cluster memory would be the most frequently used files that fit into the cluster memory. The starting point of our algorithm is the set of most frequent files that fit into the cluster memory, called MemoryFiles as shown in Fig. 3(a). Under the partition strategy, the maximum amount of data are stored in the ClusterMemory, at the price that $(N-1)/N$ of the requests coming to each node have to be handed off where N is the number of nodes in the cluster. Under the CARD strategy, all data are represented by three groups of files as shown in Fig. 4, $Files_{hot}$ and $Files_{partition}$ in the ClusterMemory, and $Files_{ondisk}$ on the disks.

Under the proposed strategy, MemoryFiles is composed such that m front-end nodes use a replication strategy, and the other (n-m) back-end nodes use a partition strategy. Then, they satisfy the following equations.

$$ClusterMemory = \sum_{front-end} Memory + \sum_{back-end} Memory \tag{3}$$



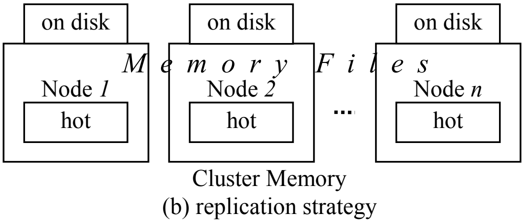


Fig. 3. Memory Files representation strategies

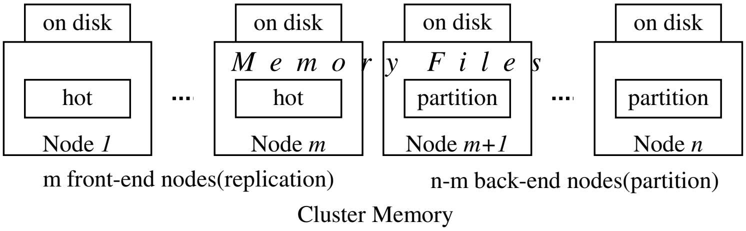


Fig. 4. Memory Files representation under new strategies

$$Files = MemoryFiles + Files_{ondisk} \tag{4}$$

$$MemoryFiles = Files_{hot} + Files_{partition} \tag{5}$$

$$m \cdot Files_{hot} \leq \sum_{front-end} Memory,\ Files_{partition} \leq \sum_{back-end} Memory \tag{6}$$

$$m \cdot Files_{hot} + Files_{partition} \leq ClusterMemory \tag{7}$$

The ideal case for Internet server request processing is when each request is processed locally, i.e. it does not incur any additional forwarding overhead, and each request is processed from memory i.e. it does not incur any additional disk access overhead. Our goal is to identify a set of hot files $Files_{hot}$ that minimizes the total overhead.

$$Workload = Workload_{hot} + Workload_{partition} + Workload_{ondisk} \tag{8}$$

First, let us analyze the overhead incurred by processing the requests to $Files_{partition}$. Assuming all these files are partitioned to be served by different nodes, statistically each file in the partition incurs forwarding overhead. The file will also incur one disk access on the node it is assigned to the first time it is read from disk. This reasoning gives us the following overhead for the partition files.

$$Workload_{partition} = \sum_{File_{partition}} (Overhead_{forward} + Overhead_{disk}) \qquad (9)$$

$$Overhead_{forward} = f \cdot T_{forward},\ Overhead_{disk} = s \cdot T_{disk} \qquad (10)$$

Here, $Overhead_{forward}$ is the processing time the TCP handoff operation consumes, and $Overhead_{disk}$ is the extra time it generally takes to read the data from disks rather than from memory. Now, consider the additional overhead incurred by processing the requests to $Files_{hot}$. If a file belongs to the hot, then the request to the file can be processed locally, i.e. with no additional forwarding overhead. The drawback is that the file has to be read from disk into memory once on all the nodes in the cluster, creating additional disk access overhead. However, assuming that the file is accessed frequently enough, it is not so serious. So as to determine which file belongs to the hot dynamically, we need to calculate the expected value of the number of nodes that get at least one access to a file given a certain frequency f and a number of nodes N.

$$E(f, N) = \sum_{i=1}^{N} i \cdot P(f, i) \qquad (11)$$

Here, $P(f, i)$ is the probability that exactly i nodes will have the file after f references to it. The overhead due to extra disk accesses and data consistency to hot files can then be calculated as follows.

$$Workload_{hot} = \sum_{Files_{hot}} (E(f, m) \cdot Overhead_{disk} + Overhead_{consistency}) \qquad (12)$$

Finally, the request to $Files_{ondisk}$ will incur additional disk overhead every time these files are accessed, which gives the following equation.

$$Workload_{ondisk} = \sum_{Files_{ondisk}} Overhead_{forward} \cdot Overhead_{disk} \qquad (13)$$

Using the reasoning and the above equations, a set $Files_{hot}$ that minimizes the total overhead can be computed. The next step is then to partition $Files_{partition}$ across the cluster. We chose to partition the files in a balanced manner using information on the load and file sizes, thus trying to put the same amount of data and load on all nodes. Given the arrival of request at a front-end node, the dispatcher goes through the following step: 1) Analyze the request. The dispatcher parses the HTTP command, and then determines whether itself or another server should server the request. 2) Distribute a request. If in hot then serve locally, otherwise handoff to back-end node.

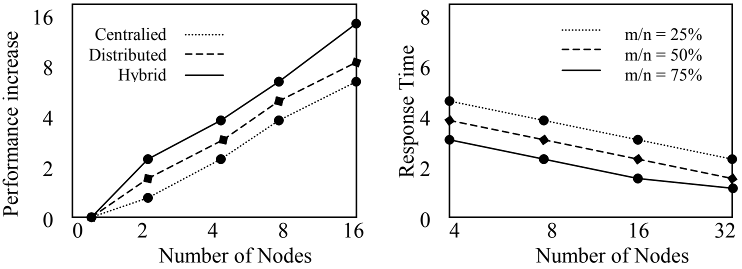


Fig. 5. Performance results under the number of node

4 Simulations

4.1 Simulation Model

The simulation model has been implemented using the CSIM18 package. The model makes the following assumptions about the capacity of the Internet server cluster: 1) Each request is executed according to the FCFS rule. 2) The cache replacement policy is LRU. 3) The cache replacement policy is LRU.

In the simulations, we used 300μsec for the forwarding overhead. The experimental results reported in [1] show that TCP handoff processing overhead is nearly 300μsec. Our goal is to assess the efficiency and performance benefits of the CARD for the current TCP handoff implementation, and to perform a hot analysis on how these results change if the number of hot is increased or decreased.

In order to run the model, the requests from the original trace were split into m sub-traces, where m is the number of front-end servers in the cluster. These sub-traces were then fed into the respective servers. Each front-end server has a finite waiting queue for requests stemming from the RR-DNS routing, while each back-end server has only a forwarded request queue. The server grabs the next request from its sub-trace as soon as it is finished with the previous request. As for the performance metrics, we measured the average server throughput across the servers after processing all the requests, and the cluster speedup achieved on the processing of the entire access log.

4.2 Simulation Results

Our first goal is to check whether our analysis identifies the hot and partition sizes close to their optimal value. In order to do this, we ran a set of simulations with varied hot sizes. Fig. 5 shows response time by increasing the number of nodes. CARD achieves linear speedup with increased cluster size. It shows excellent performance improvements compared to the traditional centralized and distributed strategy, and compares the response time to the rate of front-end node.

Table 1 shows the simulation results of the three architectures. As can be seen in the table, the hybrid approach outperforms the centralized and decentralized

Table 1. Simulation results under 3 architecture

Architecture	Utilization	Queue length	Response time
Centralized	2.90	4.32	60.15
Decentralized	2.80	4.02	57.09
Hybrid	2.71	3.70	54.47

Table 2. Simulation results under 3 architecture

Caching policy	Data access overhead time	Communication overhead time	Response time
Partition	30.6	2.0	38.36
Replication	6.54	9.0	32.71
Hybrid	6.40	5.0	32.05

approaches especially in terms of the response time. Table 2 shows the simulation results of the three caching approaches. We can see that the hybrid approach shows better response time than partition and replication approach. The data access time overhead of partition approach is worst, and the communication overhead of replication is largest. So, we can conclude that the hybrid approach has the best performance.

5 Conclusions

Researches in scalable Internet server clusters have received much attention from both industries and academia. A routing mechanism for distributing requests to individual servers in a cluster is at the heart of any server clustering technique. The scalability and the performance of an Internet server greatly depend on efficient memory usage.

We proposed a hybrid dispatcher architecture combining the centralized and decentralized approaches, and a new content-adaptive request distribution strategy which assigns a small set of most frequently used files to be served locally by the front-end servers, while partitioning the other files to be served by different cluster nodes. We also proposed an approach to compute the hot size by taking into account workload access patterns and the cluster parameters.

The simulation results showed that the proposed CARD strategy outperforms the traditional centralized/distributed strategies and a pure partitioning/replication strategy. We also investigated the idea of distributed, replicated dispatchers that use information on the previous access patterns to compute the routing information.

References

1. M. Aron, D. Snaders, P. Druschel, and W. Zwaenepoel: Scalable content-aware request distribution in cluster-based network servers, In Proceedings of the USENIX 2000 Annual Technical Conference, (2000)

2. A. Bestavros, M. Crovella, J. Liu, and D. Martin: Distributed Packet Rewriting and its Application to Scalable Web Server Architectures, in proceedings of ICN98: the 6th IEEE International Conference on Network Protocols, (1998)
3. K.L.E. Law, B. Nandy, and A. Champman: A Scalable and Distributed WWW Proxy System, Nortel Limited Reserch Report, (1997)
4. Daniel M. Dias, William Kish, Rajat Mukherjee, and Renu Tewari: A Scalable and Highly Avalable Web Server, Proceedings of IEEE COMPCON (1996)
5. L. Cherkasova: FLEX: Load balancing and management strategy for scalable web hosting service, In Proceedings of the 5th International Symposium on Computers and Communications(ISCC00), (2000) 8–13
6. V. Pai, M. Aron, M. Scendsen, P. Drushel, W. Zwaenepoel, and E. Nahum: Locality-aware request distribution in cluster-based network servers, In Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems(ASPLOS VIII), (1998) 205–216
7. Ciso Systems: Scaling the Internet Ewb Servers, A white paper available from Http://www.cisco.com/warp/public/751/lodir/scale_wp.htm, (1997)
8. X. Zhang, M. barrientos, J. Chen, and M. Seltzer: HACC: An architecture for cluster-based web servers, In Proceeding of the 3th USENIX Windows NT Synposium, (1999) 155–164
9. E.D. Katz, M. Butler, and R. McGrath: A scalable HTTP server: The NCSA prototype, In Proceedings of the First International World-Wide Web Conference. (1994)
10. Eric Anderson, David Patterson, and Eric Brewer: The MagicRouter: An application of fast packet interposing, Available from http://HTTP.CS.Berkeley.EDU/ ˜eanders/projects/magicrouter/osdi96-mr-submission.ps, (1966)
11. Jeffery Mogul: Network behavior of a busy Web server and its clients, Research Report 95/5, DEC Western Research Laboratory, (1995)

A New DSP Architecture for Correcting Errors Using Viterbi Algorithm

Sungchul Yoon, Sangwook Kim, Jaeseuk Oh, and Sungho Kang

Dept. of Electrical and Electronic Engineering, Yonsei University
132 Shinchon-Dong, Seodaemoon-Gu
120-749 Seoul, Korea
shkang@yonsei.ac.kr

Abstract. Due to the development of wireless internet and an increasing number of internet users, transferring and receiving errorless data in real-time can be the most important method to guarantee the QoS (Quality of Service) of internet. Convolutional encoding and Viterbi decoding are the widely used techniques to enhance the performance of BER (bit error rate) in the application area such as satellite communications systems. As a method to enhance the QoS of internet, a new DSP architecture that can effectively materialize the Viterbi algorithm, one of the algorithms that can correct errors during data transfer, is introduced in this paper. A new architecture and a new instruction set, which can handle the Viterbi algorithm faster, and simplify the Euclidean distance calculation, are defined. The performance assessment result shows that the proposed DSP can execute the Viterbi algorithm faster than other DSPs. Using 0.18 μm CMOS technology, the new DSP operates in 100 MHz, and consumes 218 μA/MHz.

1 Introduction

Due to the development of wireless internet and an increasing number of internet users, transferring and receiving errorless data in real-time can be the one of methods to guarantee the QoS (Quality of Service) of internet. In order to support world widely interconnected internet service, the SATCOM (satellite communications) system is used. Convolutional encoding and Viterbi decoding are the widely used techniques to enhance the performance of BER in the application area such as satellite communications system. ATM (asynchronous transfer mode) switching and IP (internet protocol), which use convolutional encoding and Viterbi decoding, can guarantee high quality channels with 10^{-8} or of higher BER[1]. Since the year 2000, demands for higher-level internet service increased. In order to accommodate such demands for faster data process, researches on SOC (system on a chip) technology, in the field for hardware infrastructure, are being increased[2-5]. Therefore, the kinds of DSP (digital signal processor), which effectively handle the Viterbi algorithm and can easily be implemented into SOC, can have considerable contributions on improvement of the QoS of internet. As a method to enhance the QoS of internet, a new DSP architecture that can effectively materialize one of the algorithms that can correct

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 95–102, 2002.

Table 1. Example of program for Viterbi decoder

```
for(i = 0; i ≤ numbit + tb_depth − 1; i + +){

        MSG_GEN( &psbuf[0], &msgbit);
        ENCODER(&msgbit, &s1, &s2, &encosrg);
        CHANNEL(&s1, &s2, &r1, &r2, &ebno);
        QUANTIZER(&r1, &r1_quant, &nbit, &fs);
        QUANTIZER(&r2, &r2_quant, &nbit, &fs);

        BRANCH_METRIC();
        PATH_METRIC();
        TRACEBACK();
        DECODED_OUTPUT();
}
```

errors during the data transfer, the Viterbi algorithm, and has a flexibility to form an SOC with an MCU and other peripherals as well, is introduced in this paper.

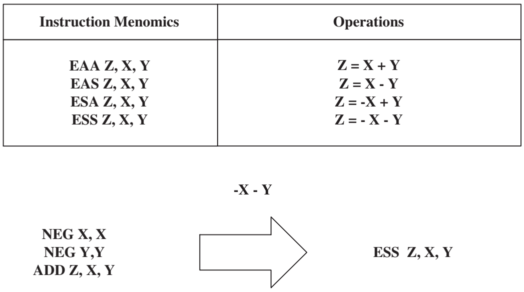
Existing DSPs are designed mainly to process SOP(sum of product)s effectively; therefore, these DSPs show good performances in applications that use a lot of SOPs, such as filter calculation. However, in some calculations, they show loss of performances due to low ILP (instruction level parallelism). Power dissipation may occur when the multiply instruction is used to calculate the Hamming distance in order to materialize the Viterbi decoding [6-7]. The proposed DSP proposes a new instruction set for the Viterbi decoding algorithm materialization, and shows a new architecture to process those instruction sets effectively. Performance evaluation shows that the proposed DSP has the most effective architecture to process the Viterbi decoding.

2 Instruction Set for Viterbi Decoding Algorithm

Table 1 shows an programming example of a typical Viterbi decoding process. With hard-decision inputs, the local distance used is the Hamming distance. This is calculated by summing the individual bit differences between the received and the expected data. With soft-decision inputs, the Euclidean distance is typically used. This is defined (for rate $1/C$) by:

$$local\_distance(j) = \sum_{n=0}^{C-1} [SD_n - G_n(j)]^2 \tag{1}$$

where SD_n are the soft-decision inputs, $G_n(j)$ are the expected inputs for each path state, j is an indicator of the path, and C is the inverse of the coding rate. This distance measure is the C^2-dimensional vector length from the received



Instruction Menomics	Operations

EAA Z, X, Y
EAS Z, X, Y
ESA Z, X, Y
ESS Z, X, Y
	Z = X + Y
	Z = X - Y
	Z = -X + Y
	Z = - X - Y

Fig. 1. Powerful ADD/SUB instructions for soft decision

data to the expected data. To minimize the accumulated distance, we are concerned with the portions of the equation that are different for each path. The terms $\sum_{n=0}^{C-1} SD_n^2$ and $\sum_{n=0}^{C-1} G_n^2(j)$ are the same for all paths, thus they can be eliminated, reducing the equation to:

$$local\_distane(j) = -2 \sum_{n=0}^{C-1} SD_n G_n(j) \tag{2}$$

Since the local distance is a negative value, its minimum value occurs when the local distance is the maximum. The leading -2 scalar is removed and the maximums are searched in the metric update procedure. For the local distance calculation like this, the proposed architecture has powerful ADD/SUB instructions, which enables users to add or sub signed numbers efficiently. Fig. 1 shows that new instructions such as EAA, EAS, ESA, and ESS can remove additional cycles for negating operations. The proposed DSP supports instructions for the ACS part, which assigns the shortest path through the Viterbi decoding process. This means that the proposed DSP supports instructions that searches for either the maximum or the minimum between two values and stores the paths into the Viterbi shift registers in sequence. MAX/MIN instructions are the instructions that can set a flag according to true/false signal, resulted from comparing two data.

The Viterbi decoding is one of the fast calculations, when several instructions get executed a loop. Each function forms a loop, and it is typical for the Viterbi function to get repeatedly executed also in a loop. It has a characteristic of using the branch instruction, to move from one loop to another, constantly. In order to use such characteristics, the proposed DSP provides loop instructions that can be nested into three levels. To minimize the loss of performance from using branch instructions many times, all instructions get processed as conditional instructions in order to be able to put into delay branch slots with no problem. The T bit is set by a compare instruction, and can be also used by a branch instruction. By placing conditional instructions in delay slots, the penalty from untaken branches can be reduced. Also, since this structure prevent fetched instructions from being issued, the power consumption becomes lower.

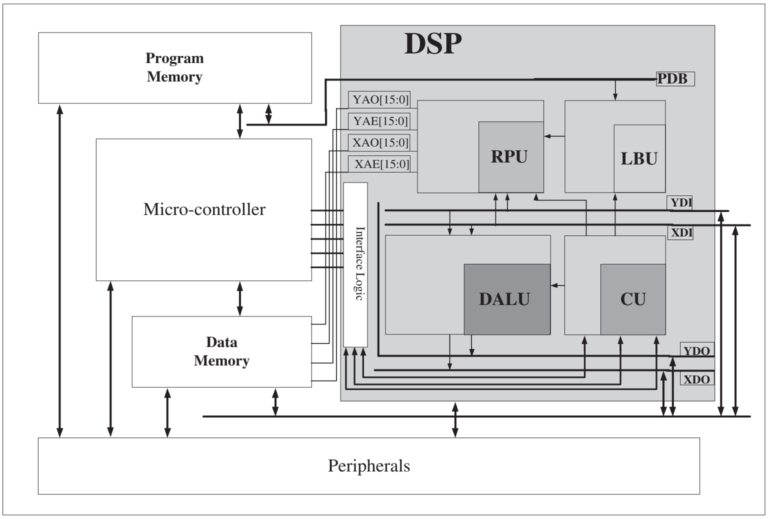


Fig. 2. A system with the proposed DSP

In EREP/EREPS instructions, the type tells whether instructions in the loop are filled in a loop buffer or not. The following example is the case of filling the loop buffer, which can be fetched from the buffer at the next turn of the loop.

EREP (typec or typed), LABEL, #16

For Viterbi decoding, EMIN or EMAX instructions can be used. There are Viterbi shift registers in DALU, VTSR0 and VTSR1. AC1, the DALU control register, contains EVS bit and ESP bit. If EVS is set, the carry of EMIN/EMAX instructions is entered into VTSR0[15] or VRSR1[15], and these registers get shifted. If ESP is set, the pointer to the minimum or to the maximum is saved. The following example shows the conditional instructions for Viterbi decoding.

(T) EMIN dest1, sour1 || EMIN dest2, sour2 || VTRSHR instruction

3 Hardware Structure for Viterbi Decoding Algorithm

The new architecture is composed of four blocks as shown in Fig. 2. Since the DSP was designed for a coprocessor, it can be used to implement a system with program memories, data memories, a microprocessor, and peripherals. LBU(loop buffer unit) fetches the current instruction, behaves as an instruction cache, and calculates branch addresses. This block helps Viterbi decoding loops to be executed rapidly. RPU(ram pointer unit) calculates two data addresses for data memory access, and is divided into two regions: X region and Y region. DALU(data arithmetic and logical unit) performs actual data calculations, and contains two MACs, BMU(bit-manipulation unit), and an internal register file.

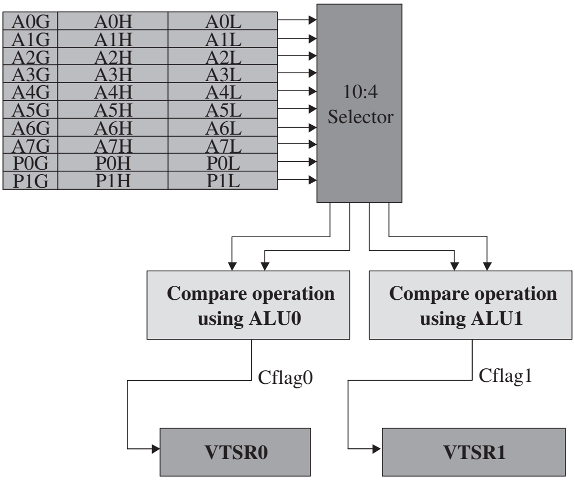


Fig. 3. Viterbi decoding paths of DALU

Finally, CU(control unit) controls the pipeline flow and bus arbitration. Since the DSP has the dual MACs, three 32-bit data buses exist for efficient memory accesses. A program bus is 32-bit wide, and four data address buses are divided into two regions of 16-bit. Fig. 3 shows the behavior of VTRSHR instruction; an instruction that does the Viterbi shifting. The DALU shown in Fig. 3 is the unit which performs either arithmetic or logical operations of data. Since two EMAX/EMIN instructions can be executed simultaneously, ACS operations of two butterflies can be performed and VTSR0/VTSR1 registers can save two path metrics independently. In addition, the LBU supports the efficient processing of loop instructions. Conditional instructions considered by T flag in a status register reduce the penalty of mis-predict branches. Fig. 4 shows the overall structure of the DALU. As shown in Fig. 4, a register file contains the registers used not only for multipliers, but also for accumulators. In addition, some accumulators share the position with the registers of multipliers. This register file has high performance in successive multiplications and additions after a multiplication. Since an input register and an output register of a multiplier are the same, successive multiplications can be executed without an additional MOVE instruction as shown in Fig. 5. This structure enables Viterbi decoding using the hard decision inputs to be executed efficiently.

4 Performance Evaluation and Implementation Result

To prove the performance of the proposed DSP in Viterbi decoding, a benchmarking with other three dual MAC DSPs has been done. The results are shown in Table 2. Although they are all dual-MAC DSPs, they are very different in detail. For example, Teak^{TM}DSP has no instruction cache and buffer, whereas

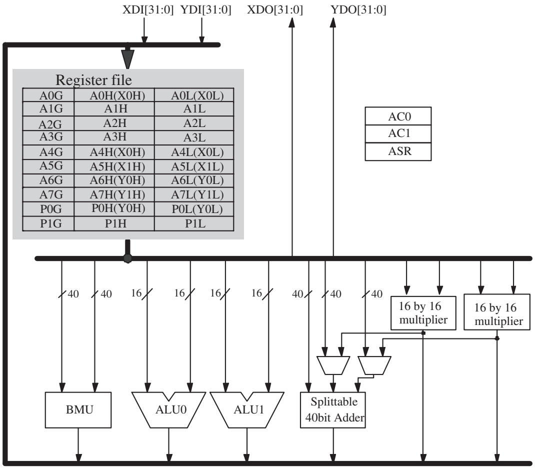


Fig. 4. DALU block diagram

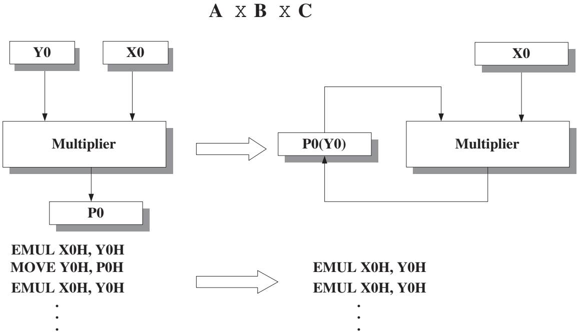


Fig. 5. Efficient structure for repetitive multiplications

TMS320C55x has both of them. Also, the number of accumulators is different among DSPs. However, all other three DSPs have the serial MAC structure to perform SOP operations efficiently. As explained previously, the MAC structure of the proposed DSP is different from other DSPs. The structure of MACs, which can reduce the number of additional negating instructions in BMG(branch metric generation), increases its performance. And two independent Viterbi shift registers having each path-metric calculation ability also makes the DSP to show the best performance. In addition, powerful ADD/SUB instructions for signed data, latency-free loop instructions, various MAX/MIN instructions, and the efficient ILP structure enhance the performance of the DSP. Finally, conditional instructions in delay slots of conditional branches reduce damages of not-taken cases by executing these instructions. Table 3 shows that the proposed DSP has the best performance of all for a Viterbi algorithm porting using code rate 1/2 and 8 delay registers.

Table 2. Architecture comparison

	Teak^{TM}DSP [6]	*DSP16000* [7][8]	*TMS320C55x* [9]	*Proposed DSP*
Data bus	three 32-bit	a 32-bit	three read 16-bit two write 16-bit	three 32-bit
Program bus	a 32-bit	a 32-bit common	a 32-bit	a 32-bit
Instruction length	16 or 32	16 or 32	8 to 48	31
Instruction cache	None	124 bytes	24 Kbytes	124 bytes
Instruction buffer	None	None	64 bytes	64 bytes
MAC	2 multipliers a 3-input ALU	2 multipliers a 2-input ALU a 3-input adder	2 MACs a 2-input ALU	2 multipliers a 3-input ALU a 2-input adders
Multiplier input	X0, X1, Y0, Y1	X, Y	Not specified	X, Y
BMU	ALU	Separated BMU	Shifter	Separated BMU
Accumulator	4	8	4	10

Table 3. Benchmark results

Procedure	*Teak^{TM}DSP*	*DSP16000*	*TMS320C55x*	*Proposed DSP*
BMG & ACS	22N	26N	22N	21N

Table 4. Implementation result

Number of gates	*Power consumption*	*Fault coverage*
59327	218 μA/MHz	93.82 %

The new DSP is implemented with 0.18μm CMOS technology library. The total number of gates is 59327, and DALU occupies 29.5% of these gates. The result is summarized in Table 4.

5 Conclusion

The new DSP architecture is a 32-bit dual MAC DSP, and behaves like a co-processor of a micro-controller. Regardless of the input type, whether it is hard decision or soft decision, the following improvements are made to achieve high performance in the Viterbi algorithm.

First, powerful ADD/SUB instructions for signed numbers is defined. Various cases due to sign combinations of two or three numbers happen in Viterbi decoding. In this case, without additional negation instructions, the DSP can execute BMG operations efficiently.

Second, an efficient 3-way super-scalar structure is used. Since each instruction can perform two operations simultaneously, it can actually be regarded as a 6-way super-scalar architecture. More ILP in the proposed DSP help higher code-rate Viterbi decodings being performed more rapidly.

Third, LBU-supported loop instructions accelerate the processing of Viterbi subroutines. Latency-free loop operations enable Viterbi subroutines to be executed rapidly.

Finally, conditional instructions reduce the mis-prediction penalty of delayed branch. In case of mis-prediction, issued instructions can be flushed by T bit. This scheme is meaningful because a Viterbi decoding program has many branch instructions.

The benchmark results show that the new DSP has the best performance in comparison with other dual MAC DSPs in Viterbi decoding. In addition, because it is a super-scalar architecture, it has an advantage in program memory usage. Therefore, the proposed DSP is suitable for the mobile technology, and since the DSP behaves as a coprocessor, it can be used as in the form of SOC.

References

1. HEISSLER, J.R., etc: An analysis of the Viterbi decoder error statistics for ATM and TCP/IP over satellite communication. Proc. Of Military Communications Conference (1999) 359–363
2. BUSS, D.D., etc: Si technology in the Internet Era. Proc. Of The 8th IEEE International Conference (2001) 1–4
3. BUSS, D.D., etc: DSP & analog SOC integration in the Internet era. Emerging Technologies Symposium: Broadband, Wireless Internet Access (2000) 5–9
4. LEE C. H., etc: Efficient random vector verification method for an embedded 32-bit RISC core. Proc. of the Second IEEE Asia Pacific Conference on ASICs (2000) 291–294
5. CHO S. Y., etc: CalmRISC^{TM} 32: a 32-bit low-power MCU core. Proc. of the Second IEEE Asia Pacific Conference on ASICs (2000) 285–289
6. Teak^{TM}DSP Core Architecture Spec. Revision 1.3
7. DSP16000 Reference Guide. (Dec 1997)
8. ALIDINA M., etc: DSP16000: a high performance, low-power dual-MAC DSP core for communications applications. Proc. of Custom Integrated Circuits Conference(CICC) (1998)119-122
9. TMS320C55x DSP CPU Reference Guide. (May 2000)

An Adapting Weight Rerouting Algorithm for Handoff Control in Wireless ATM Networks

Moonsik Kang[1], Junho Lee[2], and Sangmin Lee[1]

[1] Kangnung National University, Kangnung, Korea
sangmin@kangnung.ac.kr
[2] Seoul Nat'l University of Technology, Seoul, Korea

Abstract. A new connection rerouting algorithm for handoff control in wireless ATM networks is proposed. The proposed scheme, which we call adapting weight rerouting(AWR) algortihm, performs rerouting by choosing an optimal virtual path consisting of the network links with the optimal bandwidth available under the constraint of maximum allowable delay. The availability of network links is represented by the link weights and the optimal path is chosen based on those values. These weight values are maintained in a table, and the connection is performed by table look-up method. The proposed AWR algorithm is reasonably fast, adapts dynamically to the changing link states, and guarantees QoS by reducing the handoff blocking probability. The performance of the AWR scheme is analyzed by numerical simulation. The results indicate that the proposed scheme keeps the handoff blocking probability low.

1 Introduction

With the rapid growth of digital wireless communication services, the mobile users increasingly demand ubiquitous network connectivity in their mobile communication / computing systems. They want to access the communication services provided by the wired broadband networks such as asynchronous transfer mode(ATM) network. As an effort to meet these demands, considerable interest has recently been focused on wireless ATM, which is generally regarded as an effective and economic solution for the integration of the wireless connections in a wired broadband ATM network [1,2].

The integration of wireless networks and wired backbone ATM networks poses many technical challenges [2]. Connection rerouting for handoff is one of the important of these challenges. As mobile users move through the network while communicating, the quality of the radio link between wireless terminal and its radio access port may degrade. In this case, a new access port with an acceptable link quality must be found. The handoff is the procedure by which a mobile user's radio link is transferred from one radio port to another to avoid such degradation of the quality of communication link.

Handoff may require connection rerouting, i.e., a change in the optimal route of each active virtual channel(VC) in the wired backbone ATM network. Connection rerouting should be performed in such a way that the perceived QoS may

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 103–113, 2002.

not degenerate while the cost of the handoff is kept low. As the today's wireless networks tend to have micro/picocellular architecture in order to provide higher capacity, the rate of handoff increases and the efficient management of rerouting for handoff is crucial in providing mobility in wireless ATM networks.

The rerouting algorithms which have been proposed so far may be classified into three categories: cell forwarding, virtual tree-based, and dynamic rerouting [3]. In cell forwarding algorithm [4,5], switching function is distributed to base stations, and when handoff occurs, an add-on connection is created to handle it. Thus, this algorithm has the disadvantages that the resource is wasted due to the overlapping of previous connection with the extension connection and that it is only efficient in a flat network topology. In virtual tree-based algorithm [6,7], multiple connections to all potential destinations are created in advance and performs immediate rerouting when handoff actually occurs. Thus, this algorithm is fast, but has the disadvantage that resources are wasted since it pre-establishes connections for all potential handoffs, which may never actually occur. The NCNR algorithm [3], which is one of the dynamic rerouting algorithms, utilizes resources more efficiently compared to virtual tree-based algorithms. However its performance needs to be improved. The NCNR route is not an optimal path. If sufficient bandwidth is not available in NCNR route, then the handoff will be dropped.

In this paper, we propose a novel new rerouting scheme, called an adapting weight rerouting(AWR) algorithm, for supporting high-performance handoff in wireless ATM network. The proposed AWR algorithm performs connection rerouting by choosing an optimal virtual path consisting of the network links under the constraint of maximum allowable delay. The optimal path is chosen based on link weights, which are the values that represent the availability of network links. These link weights are maintained in a table and are updated according to the changing link states. The proposed algorithm is fast since rerouting is performed by table look-up, is dynamic since it adapts dynamically to the changing link states, and enhances QoS since the handoff blocking probability is reduced by choosing the optimal path.

The paper is organized as follows. In Section 2, the model of wireless ATM network over which handoff problem is considered is presented. In Section 3, a new rerouting algorithm for efficient handoff is presented. In Section 4, a dynamic resource reservation scheme is described, which enhances the QoS performance of the handoff when combined with the proposed AWR algorithm. Next, in Section 5, the performance of the proposed AWR algorithm is analyzed by numerical simulation. Finally, a brief conclusion is given in Section 6.

2 Handoff Procedure in Wireless ATM Networks

Wireless ATM network consists of mobile terminals, base stations, and ATM switches. The collection of base stations connected to fixed ATM network via an ATM switch is called a zone [3]. A zone is managed by a zone manager located in the zone. Two different types of handoff can be distinguished. When the mobile user is moving within a zone, the handoff is called the intra-zone handoff. When the user moves across the zones, the handoff is called the inter-zone handoff. Since the rerouting in the case of intra-zone handoff does not involve ATM network switching, we limit our attention only to inter-zone handoff.

Inter-zone handoff occurs as mobile user moves across the boundary of zones. As the user approaches the boundary, handoff is detected and managed by some process. Depending on who detects and controls the handoff, the handoff control mechanisms are classified into three types: network controlled handoff(NCHO), mobile controlled handoff(MCHO), and mobile assisted handoff(MAHO) [8]. In NCHO, the network manages the quality of the wireless channel and carries out handoff procedure if the quality of channel is under its threshold. In MCHO, the mobile terminal measures the quality of radio channel and handles the handoff. In MAHO, the mobile terminal measures the quality of the radio channel and informs the base station of the result periodically. Then the base station carries out the handoff procedure. In MAHO, the handoff calls can be directly forwarded into a candidate base station. Therefore, the signal-handling load as well as the handoff time can be reduced compared with other methods.

In this paper, MAHO mechanism is adopted and it is assumed that the handoff procedure basically consists of the following steps. 1)The mobile terminal finds that a link of better quality exists to a new candidate radio port. 2)The mobile terminal informs the present zone manager(PZM) that it desires a handoff to the candidate port. 3)The PZM establishes an ATM connection to the candidate zone manager(CZM), and sends a rerouting request with a copy of user profile to the CZM. 4)The CZM establishes ATM connection to the endpoint by performing the proposed rerouting algorithm to be described in section 3 and requests rerouting. 5)The endpoint verifies rerouting. Upon receiving this verification, the CZM assigns a channel to the user, and relays the channel assignment information to the PZM. 6)The PZM relays the channel assignment information to the mobile terminal. The mobile terminal tunes to the new channel, contacts the CZM, and verifies the connection. 7)The CZM verifies the connection to the end point, and the endpoint starts sending rerouted data to the CZM.

3 The Proposed Rerouting Algorithm

In this section, we describe the proposed adapting weight rerouting(AWR) algorithm. The AWR algorithm performs rerouting by choosing the optimal virtual path(VP) and by allocating the unused bandwidth of the chosen optimal VP to handoff call. The optimal VP is chosen based on link weights, which are the values that represent the availability of network links. These link weights are maintained in a table and are updated according to the changing link states.

Each zone manager process maintains a network link state table. The leftmost column and the topmost row of this table list all the ATM switches, say $S_1, S_2, \ldots, S_N$, which are reachable from the zone manager's node. The ij-th entry of the table is then a pair (p_{ij}, d_{ij}), where p_{ij} is the normalized weight representing the preference of choosing the link $S_i \rightarrow S_j$(i.e., the link from

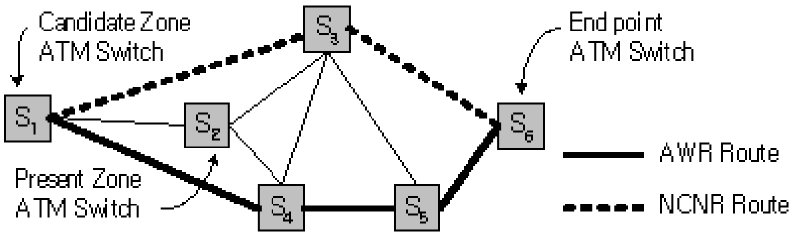


Fig. 1. A sample network of 6 wireless ATM switches

Table 1. A link state table of the zone manager of node S_1

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	-	(0.35,2)	(0.2,3)	(0.45,3)	-	-
S_2	(0.35,2)	-	(0.05,3)	(0.6,2)	-	-
S_3	(0.2,3)	(0.05,3)	-	(0.05,2)	(0.175,3)	(0.325,7)
S_4	(0.45,2)	(0.6,2)	(0.05,2)	-	(0.5,4)	-
S_5	-	-	(0.175,3)	(0.5,4)	-	(0.675,2)
S_6	-	-	(0.325,7)	-	(0.675,2)	-

switch S_i to switch S_j) and d_{ij} is the estimated delay of the link $S_i \rightarrow S_j$. At the beginning of the link state table update period, p_{ij} is set according to an estimated available bandwidth of the link $S_i \rightarrow S_j$. During the link state table update period, p_{ij} may be changed according to the result of the rerouting. The zone manager can check the states of network links and re-establishes this table periodically or whenever the network is not busy.

Using the link state table, the zone manager makes an optimal VP list, that is, a list of optimal virtual paths from the zone manager's node to all other ATM switches in the network. The optimal VP list can be made by searching the longest path with respect to the p_{ij} under the delay constraint. For example, if there are 3 candidate paths from node S_i to node S_j and the smallest normalized weights of the 3 paths are 0.5, 0.45 and 0.3 respectively, the first path is selected as the optimal path from node S_i to node S_j. The zone manager re-establishes this optimal VP list each time the network link state table is updated.

For example, in the sample network shown in Fig. 1, suppose that the present connection is between S_2 and S_6 and that the mobile terminal moves from the zone of S_2 to the zone of S_1. The zone manager of node S_1 establishes a network link state table as shown in TABLE 1. Then, if the maximum delay constraint is 9, the zone manager of the node S_1 would establish the optimal VP list as shown in Table 2. Thus, for the handoff from node S_2 to node S_1, the optimal rerouting path between the candidate node S_1 and the end point node S_6 is '$S_1 - S_4 - S_5 - S_6$'. The Fig. 1 shows this optimal AWR path by bold solid lines. For comparison, Fig. 1 also shows the NCNR path by bold dotted lines.

When handoff occurs and rerouting is needed, the zone manager looks up the optimal VP list and retrieves the optimal VP information. Then, the zone

Table 2. Optimal VP list of node S_1

destination node S_j	Optimal VP from node S_1 to S_j
S_2	S_1-S_2
S_3	S_1-S_3
S_4	S_1-S_4
S_5	S_1-S_4-S_5
S_6	S_1-S_4-S_5-S_6

manager forwards the reroute request message to all of the switches constituting the optimal VP. The nodes that receive the reroute request message check for resource availability using predefined resource management scheme. If the resources required by the connection are available, the necessary connections are established. If not, the handoff attempt fails and the involved parties are notified. When the switches on the network manage resources, a dynamic resource reservation scheme can be used to further reduce the handoff call blocking probability. Section 4 describes an efficient dynamic resource reservation scheme that can be used with the rerouting algorithm proposed in this section.

If the rerouting of data is successfully performed, then the remaining radio level handoff procedure is continued. Otherwise, the requested handoff is blocked. The zone manager updates the weight values in the network link state table according to the result of the rerouting. If the requested rerouting is successfully performed, we assume that the links constituting the rerouting path may have plenty of unused bandwidth. Thus the normalized values of these links are increased by pre-determined amount, say 5 % of the original value. If the rerouting fails, it means the links constituting the rerouting path do not have enough unused bandwidth. Thus the normalized weight values assigned to these links are decreased.

As successive rerouting occurs, a normalized weight value for any one link may be increased or decreased to the point that additional change does not represent real state of the link any more. To avoid such a situation, an upper and lower limit for a normalized weight value is predetermined. The normalized weight can be changed between the two limits.

The AWR algorithm focuses on minimizing the handoff blocking probability more than on minimizing the handoff time delay. Thus, the optimal rerouting path is determined based on normalized weights representing estimated unused bandwidth of network links. However, the normalized weight is not the actual real-time information about the unused bandwidth. The normalized weights are only estimated values collected by the zone manager. Thus we call the algorithm 'the adapting weight'. The AWR algorithm is reasonably fast since rerouting is performed by table look up. It adapts dynamically to the changes of the network link states, since the zone manager periodically updates the look-up table according to the known information about the network link states. Also, since it always allocates optimal VP for handoff connection, it reduces the handoff blocking probability.

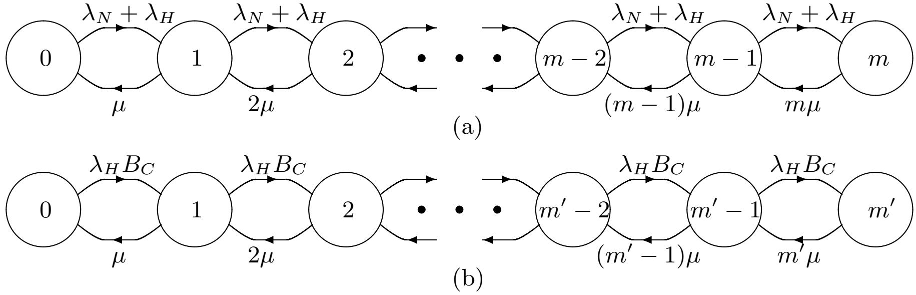


Fig. 2. State transition diagram for process $X(t)$ and $Y(t)$

4 Dynamic Resource Reservation Scheme for Connecton Rerouting

In this section, we present an efficient dynamic resource reservation scheme which can be used with the AWR algorithm to reduce the handoff blocking probability and support various QoS requirements. The scheme described here is based on the reservation scheme proposed in [9].

The total resource M of the link between any two WATM switches is divided into two portions: the common resourceM_C available for both new and handoff calls and the reserved resource M_R available only for handoff calls. The amount of M_R can be dynamically determined based on the information about network traffic condition which is periodically collected by the zone manager. When the resource is requested by a new or handoff call, resource allocation is performed as follows. If the M_C resource is available, then both new and handoff calls are accepted by using the M_C resource. If the M_C resource is not available, then new calls are blocked, and only handoff calls are accepted by using the M_R resource. If even the M_R resource is not available, then handoff calls are also blocked.

Given M_C and M_R, the new and handoff call blocking probabilities can be calculated by using a a Markov chain modeling of the system. For analysis, we assume that bandwidth is the only network resource to be considered and that all calls are of the same type requesting the same amount of bandwidth w. We also assume that the new and the handoff calls are generated according to Poisson process with rates λ_N and λ_H, respectively and that the service time of the new and handoff calls have exponential distributions with mean $1/\mu$.

Let $X(t)$ denote the number of calls being served by M_C resource at time t, and let $Y(t)$ denote the number of calls being served by M_R resources at time t. Then, $X(t)$ is an aperiodic irreducible Markov process on the state space $S = \{n \in Z | wn \leq M_C\}$, where Z denotes the set of all nonnegative integers. Similarly, $Y(t)$ is an aperiodic irreducible Markov process on the state space $S' = \{n \in Z | wn \leq M_R\}$. Fig. 2 (a) and (b) show the state transition diagrams for the time-discretized versions of the processes $X(t)$ and $Y(t)$, respectively.

The new call blocking probability can be calculated from the time-discretized version of the processes $X(t)$ shown in Fig. 2 (a). This is a $M/M/m/m$ queuing system [10], where m is the maximum integer in S. Thus, the stationary probability of $X(t)$ being in state n is

$$\pi_n = p_o \frac{\rho^n}{n!},\ n = 0, 1, 2, ..., m, \tag{1}$$

where

$$p_o = \left[\sum_{k \in S} \frac{\rho^k}{k!} \right]^{-1} \text{ and } \rho = \frac{(\lambda_N + \lambda_H)}{\mu}. \tag{2}$$

Therefore the probability that calls (either new and handoff calls) cannot receive service of M_C resource can be obtained by

$$B_c = 1 - \sum_{n \in \bar{S}} \pi_n,\ \text{where } \bar{S} = \{n \in S \mid wn \leq M_C - w\}, \tag{3}$$

and the new call blocking probability is $B_N = B_C$.

The handoff call blocking probability can be calculated similarly from the time-discretized version of the processes $Y(t)$ shown in Fig. 2 (a). This is a $M/M/m'/m'$ queuing system, where m' is the maximum integer in S'. Thus, the stationary probability of $Y(t)$ being in state n is

$$\pi_n^{'} = p_o^{'} \frac{\rho_{BH}^n}{n!},\ n = 0, 1, 2, ..., m', \tag{4}$$

where

$$p_o^{'} = \left[\sum_{k \in S'} \frac{\rho_{BH}^k}{k!} \right]^{-1} \text{ and } \rho_{BH} = \frac{(\lambda_H B_C)}{\mu}. \tag{5}$$

and thus the handoff blocking probability can be obtained by

$$B_H = \left(1 - \sum_{n \in \bar{S}'} \pi_n^{'} \right) B_C,\ \text{where } \bar{S}^{'} = \{n \in S^{'} \mid wn \leq M_R - w\} \tag{6}$$

Since this scheme reserves resource for handoff calls, it surely can reduce the handoff blocking probability B_H at the price of slight increase in new call blocking probability B_N. This is a reasonable approach because users are in general more upset when handoff call is dropped than when new call is blocked. Another advantage of this scheme is that it can adapts to dynamic change of network traffic condition by periodically modifying the amount of reserved resource M_R. The zone manager periodically collect the information about network traffic condition (for example, λ_N, λ_H, and $1/\mu$), and compute B_N and B_H. Then it compares these with the QoS requirements, and increases or decreases the amount of M_R based on the result of the comparison.

Table 3. Data assumption for numerical simulation

Traffic Type	Traffic Type 1				Traffic Type 2			
Service Type	1	2	3	4	5	6	7	8
Required Bandwidth(kbps)	32	64	256	1M	64	100	256	1M
$1/\mu$ (sec)	120	120	180	240	120	120	180	240
λ_N	0.25	0.25	0.2	0.15	0.25	0.2	0.15	0.1
λ_H	0.2	0.2	0.15	0.05	0.2	0.08	0.05	0.025

5 Performance Evaluation

In order to verify the performance of the proposed AWR algorithm, we performed numerical simulations over the sample network shown in Fig. 1, using a well-known commercial network simulaltion package SIMAN/UNIX. The traffic parameters used for numerical simulations are shown in Table 3. In simulation, it is assumed that the network is in normal state and the trunk capacity of the link between any two wireless ATM swtiches is 10Mbps. It is also assumed that the number of mobile users in each zone has a normal distribution and the user's call/handoff requests are gernerated according to Poisson process. We distinguish two types of traffic: the delay-sensitive traffic and the delay-insensitive traffic. We assume that these two types of traffic are generated with the releative frequency 6:4. We also distinguish 8 different service types. The different bandwidths required by the calls of different service types are shown in Table 3.

Simulation has been performed for 3 different cases which represent different network traffic conditions that may be encountered when the zone manager sets up network link state table. Table 4 show the network link state tables for these 3 different cases. The zone manager finds the optimal VP using the network link state table, performs rerouting using that optimal VP, and updates the table based on the result of rerouting. Successive connection rerouting would change the relevant entry values in the table up to the upper or lower threshold. In simulations, the threshold values $u_{upper} = 0.8$ and $u_{lower} = 0.2$ were used.

In simulations, we consdiered three different methods of choosing VP: AWR path(i.e., the path with the maximum weight under the delay constraint), Min-delay path and Random path. The Min-delay path can be obtained by simply searching shortest path with respect to the delay entry. The other two paths are found as follows: First, all paths which meet the delay constraint are found. Then, the AWR path is set to the one which has the maximum weight sum. The Random path is selected randomly among the candidate paths.

The results of simulations are shown in Figs. 3-5. For case 1 and 2, the maximum delay constraint of 9 was assumed, while the maximum delay of 10 was assumed for case 3. Fig. 3 shows the result of simulation for case 1. The handoff call blocking probability increases rapidly as the call arrival rate increases. Thus, when the arrival rate is high(over 2Mbps), there is no difference between the three methods. However, when the arrival rate is less than 0.5 Mbps, the handoff call blocking probability of AWR algorithm is about 30%(on the average) lower compared with those of the other schemes.

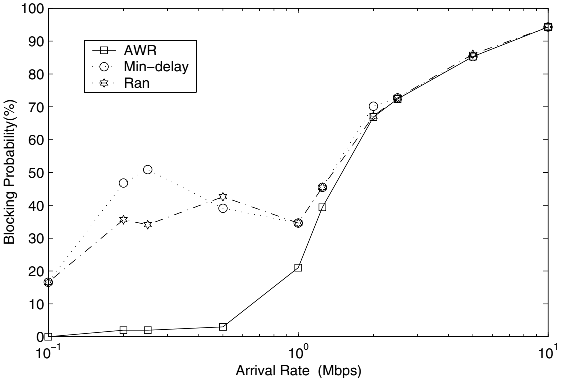


Fig. 3. Comparison of blocking Probability(CASE1)

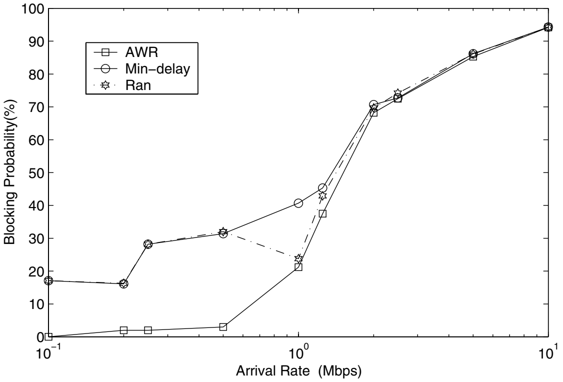


Fig. 4. Comparison of blocking Probability(CASE2)

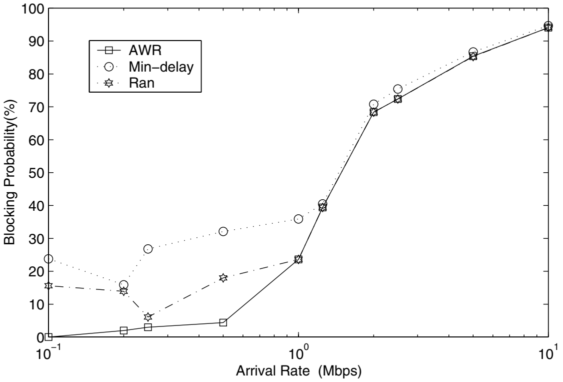


Fig. 5. Comparison of blocking Probability(CASE3)

Table 4. Network link state tables for 3 different cases

Network link state table (CASE1)

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	-	(0.45,3)	(0.45,2)	(0.1,5)	-	-
S_2	(0.45,3)	-	(0.05,3)	(0.5,2)	-	-
S_3	(0.45,2)	(0.05,3)	-	(0.05,4)	(0.05,3)	(0.4,5)
S_4	(0.1,5)	(0.5,2)	(0.05,4)	-	(0.35,2)	-
S_5	-	-	(0.05,3)	(0.35,2)	-	(0.6,2)
S_6	-	-	(0.4,5)	-	(0.6,2)	-

Network link state table (CASE2)

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	-	(0.38,2)	(0.3,3)	(0.32,3)	-	-
S_2	(0.38,2)	-	(0.26,2)	(0.36,3)	-	-
S_3	(0.0,3)	(0.26,2)	-	(0.05,4)	(0.1,2)	(0.29,4)
S_4	(0.32,3)	(0.36,2)	(0.05,3)	-	(0.28,2)	-
S_5	-	-	(0.1,2)	(0.28,2)	-	(0.71,3)
S_6	-	-	(0.29,4)	-	(0.71,3)	-

Network link state table (CASE3)

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	-	(0.55,2)	(0.0,5)	(0.45,3)	-	-
S_2	(0.55,2)	-	(0.4,3)	(0.05,2)	-	-
S_3	(0.0,5)	(0.4,3)	-	(0.05,3)	(0.05,3)	(0.5,2)
S_4	(0.45,3)	(0.05,2)	(0.05,3)	-	(0.45,2)	-
S_5	-	-	(0.05,3)	(0.45,2)	-	(0.5,2)
S_6	-	-	(0.5,2)	-	(0.5,2)	-

The simulation results for other two cases also show similar behavior. Fig. 4 shows the result of simulation for case 2. The handoff call blocking probability increases rapidly as the call arrival rate increases, thus the handoff call blocking probabilities are almost same over 1.25Mbps. This figure also shows that when the arrival rate is less than 1.25Mbps, the handoff call blocking probability of AWR scheme is about 28% lower than those of the other two schemes. This result means the AWR scheme has the enhanced performance up to the point 28%. Fig. 5 shows the result of simulation for case 3. When the arrival rate is less than 0.5 Mbps, the handoff call blocking probability of AWR scheme is about 25% lower compared with those of the other two schemes, which means that the AWR schemee has good performance. In summary, we can say that by using AWR scheme we can reduce the handoff call blocking probability remarkably when call arrival rate is low.

6 Conclusion

We have proposed a rerouting algorithm for handoff in wireless ATM network and verified its performance by numerical simulation. We have also presented an efficient dynamic resource reservation scheme, which can be used with the proposed rerouting algorithm. The proposed algorithm performs connection rerouting by choosing an optimal virtual path consisting of the network links with the optimal bandwidth available under the constraint of maximum allowable delay. The proposed algorithm is relatively fast since rerouting is performed by table look up. It adapts dynamically to the changes of the network link states, since the zone manager periodically updates the look-up table according to the known information about the network link states. Also, since it always allocates optimal VP for handoff connection, it reduces the handoff blocking probability. The performance of the proposed rerouting algorithm has been analyzed by performing numerical simulations for 4 different cases, each representing a different network traffic condition. The results indicate that the proposed algorithm keeps the handoff blocking probability very low when the arrival rate is low. The simulation results show that with the proposed algorithm we can reduce the handoff blocking probability by 20-30%.

References

1. A. S. Acampora: Wireless ATM: a perspective on issues and prospects. IEEE Personal Communications. **3-4** (1996) 8–17
2. E. Ayanoglu, K. Y. Eng, and M. J. Karol: Wireless ATM: limits, challenges, and proposals. IEEE Personal Communications. **3-4** (1996) 18–34
3. B. A. Akyol and D. C. Cox: Rerouting for handoff in a wireless ATM network. Proc. IEEE ICUPC (1996)
4. R. Yuan et al.: Mobility support in a wireless ATM network. Proc. of 5th Workshop on Third Generation Wireless Information Networks. Kluwer Publishers. (1995) 335–345
5. K. Y. Eng et al.: A wireless broadband ad-hoc ATM local-area networks. Wireless Networks. **1** (1995) 161–174
6. A. S. Acampora and M. Naghshineh: An architecture and methodology for mobile-executed handoff in cellular ATM networks. IEEE J. Select. Areas Commun. **12** (1994) 1365–1375
7. O. T. W. Yu and V. C. M. Leung: B-ISDN architectures and protocols to support wireless personal communications internetworking. Proc. of PIMRC. (1995)
8. M. Inoue, H. Morikwa, M. Mizumachi: Performance analysis of microcellular mobile communication systems. Proc. of 44th IEEE VTC. (1994) 135–139
9. K. Jang, K. Kang, J. Shim, and D. Kim: Handoff QoS guarantee on ATM-based wired/wireless integrated network. J. of KITE. **34-10** (1997) 33–51
10. D. Bertsekas and R. Gallager: Data Networks. Prentice-Hall, New Jersey (1987) 134–140

Optimal Buffering Strategy for Streaming Service in Time Varying Wireless Environment

Taechul Hong and Youngyong Kim

Dept. of Electrical and Electronics Engineering, Yonsei University, Shinchon-Dong,
Seodaemoon-ku, Seoul 120-749, Korea
{taechori,y2k}@yonsei.ac.kr

Abstract. As 3G services begins and wireless LAN gets widespread, we expect streaming service will become one of the dominant services in wireless environment. However, streaming service in wireless environment is much difficult to support than in wired network, because available rate changes according to channel situation. Therefore, streaming service quality is not guaranteed just by using delay jitter buffering. On the other side, maximum rate buffering in each state incurs expensive price. We approach optimal buffering problem for guaranteed streaming service, using utility function [1] for cost and degradation. Finally, we get the optimal buffering value by analyzing utility function and validate the model by applying to practical system such as HDR [2].

1 Introduction

Streaming service[3] in wireless environment has lately attracted considerable attention. Especially development of wireless LAN and the dawn of 3G services bring about people's expectation of mobile streaming service. However, it's hard to achieve guaranteed service quality for mobile streaming service due to time varying wireless channel characteristics. In other words, wireless environment often causes change of allowed rate according to channel situation, so we can't guarantee sustained streaming rate.

In order to ensure QoS in wireless streaming service, it is necessary to have a sufficient amount of buffering. Judging from the above facts, size of buffer must be greater than that of wired counterpart which requires the size of delay jitter bound. In addition to that, low cost is needed for successful deployment of wireless streaming services[4]. If cost of service is too high, people will be very slow to adopt the new service.

Hence, it is evident that cost factor and size of buffer are important factors for guaranteeing wireless streaming service. To take these into consideration, we define utility function which is concerned with cost and degradation. In this paper, we propose optimal buffering method for guaranteed streaming service quality, which maximize the utility of user.

The remainder of this paper is composed as follows. Section 2 describes two-state model of utility function. In Section 3, two-state model is expanded to multi-state model. In Section 4, our model is applied to practical system (HDR). Conclusions are given in Section 5.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 115–123, 2002.

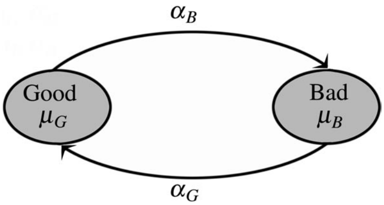


Fig. 1. Two-State Modeling

2 Two-State-Model

2.1 Streaming Rate

(α_B,α_G: state transition rate, μ_G: transmission rate at good channel, μ_B: transmission rate at bad channel)

In this section, we assume that wireless channel can be modeled by simple two states Markov Chain (see Fig. 1) and assume user wants to get streaming rate which lies between the transmission rate of good and bad channel. We denote requested streaming rate as μ_T. From above model average channel rate can be calculated as $\overline{\mu} = \pi_G \mu G + \pi B \mu B$. (where μ_G: steady state probability for the good channel, μ_B: steady state probability for the bad channel) Considering buffering mechanism, we can classify into the following cases.

$*\mu_T > \overline{\mu}$: cannot guarantee the requested streaming service rate

$*\mu_T < \overline{\mu}$: Two Cases
(1)$\mu_T < \mu_B$: Complete guarantee
(2)$\mu_B < \mu_T < \overline{\mu}$: Need to compensate for degradation

In the next subsection, we the case of compensating for the degradation since other cases are very trivial.

2.2 Degradation

As shown in Fig. 2, channel is continuously time varying between good and bad state. In good channel, supporting rate is greater than streaming rate, so degradation does not occur. However, in case of bad channel, supporting rate is lower than streaming rate, resulting in the quality degradation. This degradation is expressed as

$$\begin{aligned} Degradation &= \lim_{T\to\infty} \frac{1}{T}(T_{B_1}(\mu_T - \mu_{B_1}) + T_{B_2}(\mu_T - \mu_{B_2}) \\ &\quad + T_{B_3}(\mu_T - \mu_{B_3}) + \cdots) \\ &= (\mu_T - \mu_B)[T_{B_1} + T_{B_2} + T_{B_3} + \cdots] \\ &= (\mu_T - \mu_B)\pi_B \end{aligned} \tag{1}$$

(T_G: time duration of good channel, T_B: time duration of bad channel)

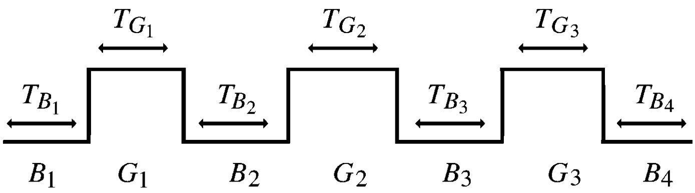


Fig. 2. Channel State According to Time

In good channel, degradation is compensated through buffering, so total degradation can be expressed as follows.

$$\begin{aligned} TotalDegradation &= \lim_{T\to\infty}\frac{1}{T}(T_{B_1}(\mu_T-\mu_{B_1}) \\ &\quad +[T_{B_2}(\mu_T-\mu_{B_2})-T_G(\mu_{G_1}-\mu_T)] \\ &\quad +T_{B_2}(\mu_T-\mu_{B_3})-T_{G_2}(\mu_{G_2}-\mu_T)]+\cdots \\ &= \lim_{T\to\infty}\frac{1}{T}[\sum_i T_{B_i}(\mu_T-\mu_{B_i})-\sum_j T_{G_j}(\mu_{G_j}-\mu_T)] \\ &= \pi_B(\mu_T-\mu_B)-\pi_G(\mu_G-\mu_T) \end{aligned} \tag{2}$$

If streaming rate is lower than average supporting rate, value of total degradation will have negative value.

2.3 Utility Function

Whenever ISP(Internet Service Provider) serves streaming contents with full rate, not every user can be satisfied. In other words, users want to get the service not only of high quality but also with low cost. However, cost increases in proportion to the rate increase in general. Therefore, optimization between cost and quality is required. In this paper, we use utility function as objective for the optimization. Utility function is widely used in economics. Here, we use degradation and cost as the parameters of utility function. We define utility function used in this paper as follows, and will explain each component later.

$$U = F(C,D) = U_C(C)U_D(D) \tag{3}$$

(D: Degradation, C: Cost)

$U_C(C)$. Depending on the various environments, a wide variety of cost policies can be applied[5]. In this paper, we assume simple cost policy which increases in proportion to the service rate.

The motivation to use this type of utility function is as follows. In general, people are discouraged to use the service as a charge increases. Therefore, cost is in inverse proportion to utility[6]. If we set maximum value of utility to one, utility function of cost is expressed as

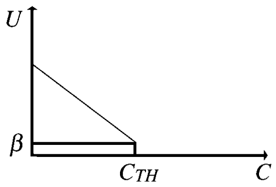


Fig. 3. Utility Function for Cost Part

$$U_C(C) = [1 - \frac{C}{C_{TH}} + \beta], \tag{4}$$

where CTH represents the cost of maximum rate service and β represents the basic utility. Basic utility means that people who are willing to pay the maximum charge for service exist. Fig. 3 shows graph for utility function for the cost part.

In bad channel, rate selection for compensation does not happen since the requested rate is higher than the transmission rate in bad channel. However, in good channel, rate selection for compensation is required when we consider the cost. We denote the actual transmission rate in good channel as x. We can represent cost when using rate x as follows,

$$C(Cost) = \alpha(\pi_B \mu_B + \pi_G x) \tag{5}$$

where α is a proportional constant. If we apply (5) to (4), we get

$$U_C(x) = (1 - \frac{\alpha(\pi_B \mu_B + \pi_G x)}{C_{TH}} + \beta) \tag{6}$$

Then (6) represents utility function for the cost part when using rate x at good channel state.

***$U_D(D)$*.** Streaming service is very sensitive to bandwidth and delay. Therefore we assume that degradation is partially elastic[6]. If degradation is larger than D_{TH} (degradation threshold), users have no utility. In other words, case of low bandwidth has no utility. However, if degradation is lower than DTH, utility is in inverse proportion to degradation. So we can express utility function of degradation as

$$U_D(D) = [1 - \frac{D}{D_{TH}}] \tag{7}$$

We can also represent utility function of degradation through variable x. Inverse proportion exists in relation between service rate and degradation as well as relation between utility and degradation. Therefore, we make degradation model with rate x, and then we apply it to (7).

In Fig. 4(a) utility function for the degradation part is shown. We translate rate into degradation in Fig. 4(b). In Fig. 4(b), D0 is a quantity of degradation in bad channel state, xT is rate required to compensate for all degradation

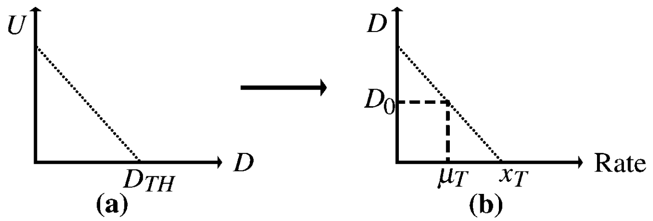


Fig. 4. (a) Utility Function for the Degradation part (b) Degradation vs. Rate Change

accumulated in bad channel state. Now, we can express degradation in terms of rate x as follows.

$$D = \frac{D_0}{\mu_T - x_T}(x - x_T) \tag{8}$$

Now, we apply (8) to (7), and finally get utility function for the degradation when using rate x,

$$U_D(x) = (1 - \frac{D_0(x - x_T)}{D_{TH}(\mu_T - x_T)}) \tag{9}$$

Optimal Buffering Rate. By combining two utility function from the previous sub-sections, we can get complete utility function through applying (6) and (9) to (3).

$$U(x) = U_D(x)U_C(x) = (1 - \frac{D_0(x - x_T)}{D_{TH}(\mu_T - x_T)})(1 - \frac{\alpha(\pi_B\mu_B + \pi_G x)}{C_{TH}}) \tag{10}$$

Finally, we get optimal buffering rate from the derivative of utility function (10),

$$\frac{dU(x)}{dx} = 2AB\pi_G x + (AB\pi_B\mu_B - AB\pi_G x_T - B\pi_G - A)$$
$$x_{opt} = \frac{-(AB\pi_B\mu_B - AB\pi_G x_T - B\pi_G - A)}{2AB\pi_G}$$
$$(A = \frac{D_0}{D_{TH}(\mu_T - x_T)}, B = \frac{\alpha}{C_{TH}})$$

and this rate will maximize user's utility.

3 Multi-state Model

We extend two-state model used in the previous model to multi-state model. Multi-state model is better to describe realistic situation. Figure 5 shows multi-state model.

3.1 Degradation

λ_i, μ_i represents for upward and downward state transition rate respectively and S_i denotes for the data rate in state i. We express S_T as streaming rate or target rate.

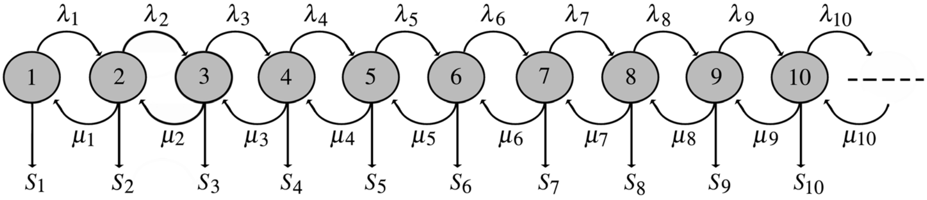


Fig. 5. Multi-State Model

The notion of degradation concept in two-state model can be applied to multi-state model as well. Degradation and total degradation are expressed as follows.

$$Degradation(D_0) = \sum_{S_i \leq S_T} (S_T - S_i)\pi_i$$

$$TotalDegradation = \sum_{S_i \leq S_T} (S_T - S_i)\pi_i - \sum_{S_i \geq S_T} (S_i - S_T)\pi_i$$

3.2 Utility Function

***$U_C(C)$*.** Multi-state model has several good and bad states, in contrast to two-state model. So cost function and utility function of cost can be expressed using summation.

$$C(Cost) = \alpha[\sum_{S_i \leq S_T} S_i\pi_i + \sum_{S_i \geq S_T} x_i\pi_i] \tag{11}$$

$$U_C(\bar{x}) = [1 - \frac{\alpha[\sum_{S_i \leq S_T} S_i\pi_i + \sum_{S_i \geq S_T} x_i\pi_i]}{C_{TH}} + \beta] \tag{12}$$

We use the notions of “good” and “bad” as meaning of states with the transmission rate greater than S_T and states with the transmission rate smaller than S_T, which is requested target rate.

***$U_D(D)$*.** Degradation in multi-state model is expressed with degradation(D_0) and compensation. Therefore, utility function of degradation is expressed through equation (7).

$$D = D_0 - \sum_{S_i \geq S_T} (x_i - S_T)\pi_i \tag{13}$$

$$U_D(\bar{x}) = [1 - \frac{D_0 - \sum_{S_i \geq S_T}(x_i - S_T)\pi_i}{D_{TH}}] \tag{14}$$

Optimal Buffering Rate. We get complete utility function for multi-state model by applying (12) and (14) to (3).

$$U(\bar{x}) = [1 - \frac{\alpha[\sum_{S_i \leq S_T} S_i\pi_i + \sum_{S_i \geq S_T} x_i\pi_i]}{C_{TH}}][1 - \frac{D_0 - \sum S_i \geq S_T(x_i - S_T)\pi_i}{D_{TH}}] \tag{15}$$

If we rewrite equation (15),

$$\begin{aligned} U(\overrightarrow{x}) &= D(\overrightarrow{x})C(\overrightarrow{x}) = [\textstyle\sum_i a_i x_i + C_1][\sum_i b_i x_i + C_2] \\ a_i &= -\pi_i/C_{TH}, C_1 = 1 - \textstyle\sum_{S_i \leq S_T} \frac{S_i\pi_i}{C_{TH}} + \beta \\ b_i &= \pi_i/D_{TH}, C_2 = 1 - \frac{D_0}{D_{TH}} - \textstyle\sum_{S_i \geq S_T} \frac{\pi i S_T}{D_{TH}} \end{aligned}$$

then we get the partial derivative of each x_i.

$$\begin{aligned} \partial U/\partial x_j &= a_j[sum_i b_i x_i + C_2] + b_j[sum_i a_i x_i + C_1] \\ &= \sum_i [a_j b_i + b_j a_i] + a_j C_2 + b_j C_1 \end{aligned}$$

Finally, we can get optimal buffering rate for maximizing utility, by solving simultaneous equation after arranging $\partial U/\partial x_j = 0$ for each j. Solution is represented by vector (equation (16)) with optimal buffering rate x_i. for each state i.

$$\bar{x}_{opt} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_N) \tag{16}$$

4 A Practical Consideration (A Case of HDR)

HDR(High Data Rate) in CDMA2000 has eleven discrete states and associated rates in each state[2]. Therefore, we apply multi-state model to HDR. (see Fig. 6)

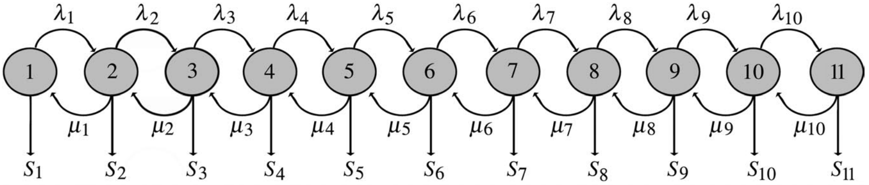
When HDR system is applied to utility function of our multi-state model, we can get equation (17).

$$\begin{aligned} U(x) = (&0.844 - 0.0003x_7 - 0.000157x_8 - 0.000217x_9 \\ &-0.0001x_{10} - 0.00000183x_{11}) \\ \times(&0.003x_7 + 0.00157x_8 + 0.00217x_9 + 0.001x_{10} \\ &+0.000083x_{11} - 3.589) \end{aligned} \tag{17}$$

(Streaming rate= 400kbps, C_{TH}=600 , D_{TH}=60)

In this case, our approach (3.2 optimal buffering rate) for buffering rate poses a limitation. Optimal buffering rate through utility function may be different from rate possible in each state. Therefore our solution does not apply to maximization of utility.

In HDR, eleven discrete states have eleven discrete supporting rate so utility optimization problem is an integer valued optimization problem. Therefore, optimization problem for utility function does not have compact algorithm and



	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
Rate (Kbps)	38.4	76.8	102.6	153.6	204.8	307.2	614.4	921.6	1228.8	1843.2	2457.6

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
Rate	0.995	0.99	0.02	0.99	0.40	0.75	0.005	0.99	0.08	0.10

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
Rate	0.01	0.25	0.01	0.53	0.25	0.995	0.01	0.74	0.20	1.0

	x_1	x_2	x_3	π_4	π_5	x_6	x_7	π_8	π_9	π_{10}	π_{11}
Probability	0.001	0.01	0.04	0.08	0.15	0.25	0.18	0.094	0.13	0.06	0.005

Fig. 6. HDR Model

in worst case, we may resort to the brute search method. In the case of HDR, small number of states makes it possible to use brute search method. However, if large numbers of states exist, one might find difficulty in solving the optimization problem.

We apply two methods to equation (17). First method is a maximum rate buffering which uses practicable maximum rate in each state without any policy. Second one is an optimal rate buffering which uses rate calculated through brute search method, for each state. The results are shown in Table 1.

Table 1. Value of Utility Function

	Maximum Rate Buffering	Optimal Rate Buffering
Value of Utility Function	0.194	0.5767

The result clearly shows that optimal rate buffering scheme increases user's utility than no-policy. Therefore, it is reasonable to conclude that streaming service in wireless environment needs to adapt streaming rate according to characteristics of time varying channel for user's satisfaction.

5 Conclusion

Buffering for streaming service is imperative to support quality guarantee. In this paper, we propose the notion of optimal solution for buffering rate through maximizing utility function. However, our solution does not apply directly to HDR-like system, which makes us to solve integer valued optimization problem through brute search method. The critical contribution here is the introduction of utility functions for the service guarantee in wireless environments. In future, we will study rate bounded utility function and analytic solution for discrete value of rate.

References

1. A. Mas-Colell: Microeconomics. Oxford Univ. Press 1995
2. Patrick A. Hosein: Capacity Model for the CDMA/HDR High-Speed Wireless Data Service. Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems. (Jul. 2001) 37–44
3. Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, Jon M. Peha: Streaming Video over the Internet: Approaches and Directions, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, **11** (Mar. 2001) 282–300
4. Toru Otsu, Ichiro Okajima, Nrumi Umeda, Yasushi Yamao: Network Architecture for Mobile Communications Systems Beyond IMT-2000.IEEE Personal Communications. (Oct. 2001) 31–37
5. Matthias Falkner, Michael Devetsikiotis, Ioannis Lambadaris: An Overview of Pricing Concepts for Broadband IP Networks. IEEE Communications Surveys. Second Quarter 2000.
6. Luiz A. Dasilva: Pricing For QoS-Enabled Networks: A Survey, IEEE Communications Surveys, Second Quarter 2000.

Optimum Ship Routing and It's Implementation on the Web

Heeyong Lee[1], Gilyoung Kong[2], Sihwa Kim[3],
Cholseong Kim[4], and Jaechul Lee[5]

[1] MobileNet Co., LTD, Pusan, Korea
jimcarry@nmobile.net
[2] Division of ship operation systems engineering
Korea maritime University, Pusan, Korea
[3] Division of maritime tranportation science
Korea maritime University, Pusan, Korea
[4] Institute of Maritime Industry
Korea maritime University, Pusan, Korea
[5] ITEP, Information and Telecommunication Team, Seoul, Korea

Abstract. Optimum Ship Routing can be defined as "The selection of an optimum track for a transoceanic crossing by the application of long-range predictions of wind, waves and currents to the knowledge of how the routed vessel reacts to these variables". Generally, Optimum Ship Routing was held prior to sail with predicted weather data, but nowadays, with rapid development of Internet/Intranet technology, it becomes possible to carry out real time weather data. This paper treats the methodology how to solve optimum ship routing problem by network modeling and shows a basic concept of the Web-Based Optimum Ship Routing System.

1 Introduction

Optimum Ship Routing can be defined as "The selection of an optimum track for a transoceanic crossing by the application of long-range predictions of wind, waves and currents to the knowledge of how the routed vessel reacts to these variables" [1].

The primary goal of ship routing is to reduce a voyage cost in various aspects and keep safe during the period of vessel underway. From ancient times, a captain has been selecting the best route considering the weather characteristics such as prevailing wave, wind and current status in specific season and area. With insufficient weather data, it is difficult to get the suitable route to save voyage cost.

Recently, the Internet becomes a prevalent infrastructure to send, receive and share information in many organizations. In this paper the optimum ship routing problem is formulated as a network model and solved by a modified depth-first search algorithm.

And this paper also shows a basic system configuration of web-based optimum ship routing system and gives explanations of weather data transferring.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 125–136, 2002.

2 Optimum Ship Routing Problem

On a spherical surface of the earth, the shortest path from a departure point (X_s,Y_s) to destination point (X_d,Y_d) is to sail along the Great Circle connecting two points. If there exist heavy seas around a Great Circle route, to sail along the route on a calm sea can save voyage time. The optimum route connecting two points is to be determined by the function of the extent of obstruction and a distance between two points.

A determination of an optimum ship route is to select the best route among a number of candidate routes, which minimize a cost such as sailing time, fuel consumption, etc considering weather condition. Decision variables of the problem are to be a control of course and speed. The kinematics of ship under sailing can be described as the function of a time and position [2].

The voyage cost is to be determined by ship's position P, control variable for engine power and course C , and time t. The port cost B is to be determined by arrival time t_f. With these components of a cost, the ship routing problem can be described by the formulation.

Minimize:

$$I = \int_s F\left(P, C, t\right) ds + \omega B\left(t_f\right) \tag{1}$$

$F(P, C, t)$: A function of voyage cost for position, control and time

t_f: Arrival time at a destination port

$B(t_f)$: A function of port cost for arrival time

ω: A weight for port cost to be reflect in voyage cost

Where, $P \in R, C \in C_A$

Regarding Ship Position P, Control Variable C as a function of time t, then the optimum control $C^*(t)$ that minimizes cost I determines the optimum route $P^*(t)$. A cost per unit distance ds is measured by sailing time, fuel consumption and the degree of ship safety under sailing. Where ω is 0, it means that a port cost is discarded, otherwise ω is regarded as a penalty term in the objective function. Let R as a possible sailing region, then control variable C_A will be a limit of allowable engine power and an extent of course changing.

3 Literature Survey

The ship routing problem can be categorized into three types according to the solution algorithm, 1) Isochrone Method, 2) N-Stage Dynamic Programming, 3) Calculus of Variation.

Among the recent researches, Hagiwara's one that suggested a modified isochrone method presented a remarkable result suitable to utilize a high computer technology[3].

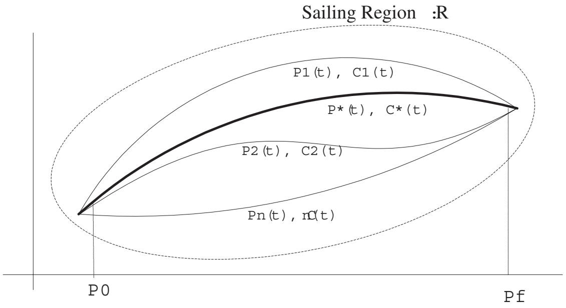


Fig. 1. Optimum route in the sailing region

3.1 Isochrone Method

An Isochrone Line (Time-Front) is defined as outer boundary of the attainable region from the departure point after a certain time. If the weather condition is constant, Isochrone Line will be a part of circle-arc.

The Isochrone Methods proposed by Hanssen G.L and R.W.James [1] had been used for a long time because it offered an ease manual calculation methods and Hagiwara devised a modified Isochrone Methods suitable for computer programming. The algorithm of Isochrone Methods is as follows.

3.2 N-Stage Dynamic Programming

N-Stage Dynamic Programming uses a Grid System, which divides a possible sailing region into several cells. Each crossing points of cell boundary are candidate of waypoints. The solution algorithm of Dynamic Programming seek ship's trajectory that is composed of position $(X, Y, T)_k$, $k = 1, 2, ..., N$, and Control C_k with initial time T_0 and initial ship position.

3.3 Calculus of Variation

Generally, calculus of variation method is applied to a problem to minimize a integral J. Let's consider a simple formulation.

Minimize:

$$J = \int_{x_1}^{x_2} f(y, y_x, x)dx \tag{2}$$

This problem seeks a function y_x to minimize integral J from x_1 to x_2. In ship routing problem, f is a cost function defined by ship position and time. The calculation result of the function varies from the condition of sea state. The optimum ship route problem solved by method of calculus of variation is to find y_x which minimize J [4][5][6].

4 Network Model

4.1 Network Generation

A network is a special form of a graph that consists of nodes and arcs. A network has numerical parameter such as the length, cost and transit time.

In this paper, a network called "Waypoint Network" is generated on the base of great circle sailing method to make a practical routes.

Nodes Generation. Nodes in waypoint network are generated with the points that lie on a great circle route and it's neighborhood routes.

To make nodes, at first draw a great circle route between two points. and seek the center point of great circle route. Then around the center point, draw adjacent point around center point in a certain distance. The set that consists of a center point and its adjacent points are called *"A set of center points"*. Connecting departure point and the point in the *set of center points* composes a half great circle routes. Similarly, connecting the point in the *set of center points* and arrival point composes a half great circle routes. An algorithm to nodes generation is summarized as follows.

Step 1. Seek center point C on the great circle route.
Step 2. Seek course δ between Dep. Point $X_s(Lat_1, Lon_1)$ and Arr. Point $X_f(Lat_2,Lon_2)$
Step 3. Draw a perpendicular δ' to δ , where $\delta' = \delta - 90$ or $\delta' = \delta + 90$
Step 4. Draw the point $c_{\pm i}$ apart from center point in certain distance ζ_i to direction of δ'. Where, $\zeta_{i+1} = \zeta_i + \zeta$
Step 5. C and $c_{\pm i}$ is a *set of center points.*

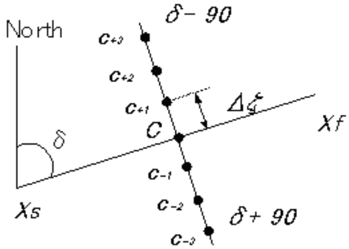


Fig. 2. Set of center points

Course δ is calculated by the formulation.

$$\delta = D_{RL}/l$$

Where, D_{RL} is Rhumb Line distance, $l = Lat_2 \sim Lat_1$: displacement of latitude.

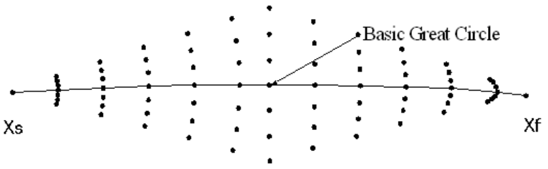


Fig. 3. Nodes in waypoint network

Initial Course A_z of great circle route is calculated by the formulation.

$$havA_z = secL_1scsD[havcoL_2 - hav(D \sim coL_1)]$$

where, $havA = sin^2(A/2)$, A_z: Initial course, L_1: lat. of dep. point, coL_2: co Lat of dep. point, D: Distance.

Distance D is calculated by the formulation.

$$D = sin^2Lat_1sinLat_2 + cosLat_1cosLat_2cosDL_0$$

The point apart from the dep. point in half distance of great circle distance is calculated by the formulation.

$$\begin{aligned} sin(D_{gc}/2) = {} & [sin^2(Lat_2 - Lat_1)/2 \\ & +cos(Lat_1)cos(Lat_2)sin^2[(Lon_2 - Lon_1)/2]]^{1/2} \end{aligned}$$

And a location (Lat ϕ, Long λ) of a point from a departure point with a certain distance g and initial course A_z is calculated by a formulation.

$$\begin{aligned} \phi &= arcsin(sin\phi_1cosc + cos\phi_1sinccosA_Z), \\ \lambda &= \lambda_0 + arctan[sincsinA_z/(sin\phi_1sinccosA_z)] \end{aligned}$$

Fig. 3 shows nodes generated by the methods mentioned above. The nodes in waypoint network spread around the basic great circle route so can prune unnecessary waypoints compared with the grid system in dynamic programming method.

Arcs Generation. Let n_s as a start point, and n_i, i / ~~s~~as the candidate point for next arrival. The course between n_s and n_i, i / ~~s~~ is δ_i, the distance is d_i. The limit of course changing is Θ.

The condition for n_i to be a next arrival point is $d_i <= D$, $\delta_i <= \Theta$. (Fig. 4)

4.2 Solution

Objective Function. The objective function of network model is to minimize sailing time, fuel consumption, and to keep more safe navigation. To produce a practical route, it is important to apply the real aspect of weather condition.

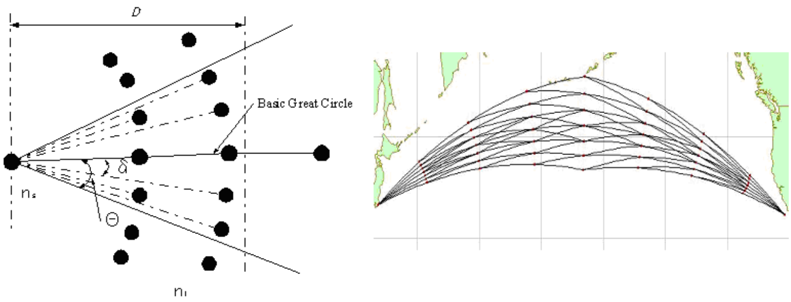


Fig. 4. Generation of arcs and waypoint network

But in this network model, only the wave and current is to be considered. That is, only the sailing speed due to weather disturbance and port cost according to arrival time is to be regarded as components of cost estimation. The Objective Function is to

Minimize:

$$J = \sum_{i \in I} T_i + C_P^T \tag{3}$$

T_i: Sailing time from waypoint i-1 to i
C_P^T: Port cost due to arrival time T
Where, I is a set of all waypoints in selected route.

Decision variables in ship routing problem are the controls of course and speed but in the network model, to decrease the size of problem, a speed is suggested to be constant and only the course is taken into account as a decision factor.

Calculation of a Cost. To make a numerical experiment simple, the amount of speed loss is calculated only by the effect of wave and current. The wind effect is ignored because its relationship with wave is in linear.

1) Estimation of speed loss
The speed loss due to wave is affected by height, direction. Head seas, beam seas and following seas are considered to estimate speed loss. Current is considered to be a head current and a stern current. The estimated speed is v' considering speed loss due to wave (v_w) and current (v_c) against a normal speed v is calculated by the formulation:

$$v' = v - (v_w + v_c)$$

To apply weather effect into speed estimation, the possible sailing region is divided into square cells in discrete manner called a weather cell. The weather cell is a region, which has a constant weather condition. Usually, the course line of two waypoints (n_s, n_i) lies on several cells. To calculate voyage cost exactly, it is required to consider the sailing line as a composition of line segments cut bon

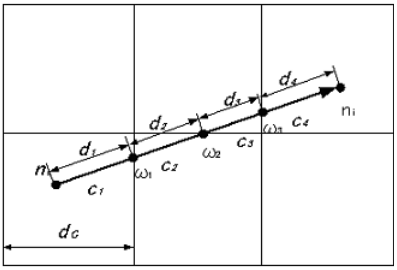


Fig. 5. Calculation of cost on multi cell

several cells. To calculate voyage cost exactly, it is required to consider the sailing line as a composition of line segments cut by a crossing points $(\omega_1, \omega_2, ..., \omega_n)$ with a cell boundary. The Fig. 5 shows a schematic diagram of line segmentation.

The sailing time between n_s and n_i is calculated by the formulation.

$$T_i = T_s + \sum_{j \in J} \frac{d_j}{v - (\zeta_i + \xi_i)} \tag{4}$$

T_s: A sailing time (cost) form dep. point to waypoint n_s

J: A set of crossing points d_j: Distance to ω_j v: A normal speed without weather disturbance ζ_i: The amount of speed loss due to current ξ_i: The amount of speed loss due to wave

Cost function $d_j / (v - (\zeta_i + \xi_i))$ means a sailing time. Total sailing time T_f to arrival point n_f is calculated by the formulation $T_f = \sum_{i \in I} T_i$, where I is a set of waypoints on a route. The port cost at the port X_f considering an arrival time T_f is $B(X_f, T_f)$ then total cost is calculated by the formulation:

$$J = \sum_{i \in I} T_i + B(X_f, T_f) \tag{5}$$

2) Port cost
The port cost is regarded to be a function for the arrival time. It is assumed that the early and delayed arrival time occurs at the same cost.

Algorithm. To pick out candidate routes from the waypoint network, a modified depth-first search algorithm is used. The modified depth-first search algorithm is to add one step more into original algorithm and use additional stack structure to store generated candidate routes.

During the traverse sequence, if the visited node is an arrival point, the path is registered as one candidate route. The gray node is arrival point and Fig. 6 shows that 5 candidate routes are generated.

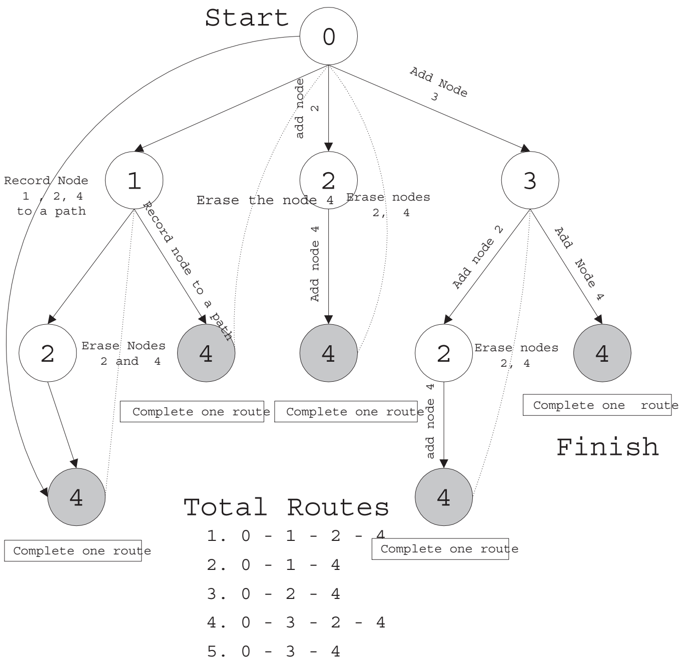


Fig. 6. Depth-First search algorithm

5 Web Based Optimum Ship Routing System

5.1 System Design

Prevalent Technologies to share computing resource under internet/intranet environment are HTTP/CGI, socket programming, Java Applet in addition to ODBC, JDBC, CORBA and DCOM. Web based Optimum Ship Routing System (WOSRS) used HTTP, Socket Programming and ODBC technology as shown in Fig. 7.

There exists Weather Information Module (WIM), Interactive Modeling Module (IMM) and Optimization Module (OM) in WOSRS client system and WOSRS client system keep interfacing to various navigation equipments including ECDIS and VDR under intranet environment. WIM can browse weather information on general Web Browser such as Microsoft Internet Explorer and Netscape Navigator. And WIM can directly connect to Weather Server on shore side using socket-programming technology. The sequence diagram for socket programming technology is shown at Fig. 8.

IMM and OM share optimum route information with ECDIS and VDR using ODBC technology. WOSRS uses ADO as local database instead of using enterprise DBMS such as Oracle, Sybase, etc.

*ECDIS: Electronic Chart Display and Information System

*VDR: Voyage Data Recorder

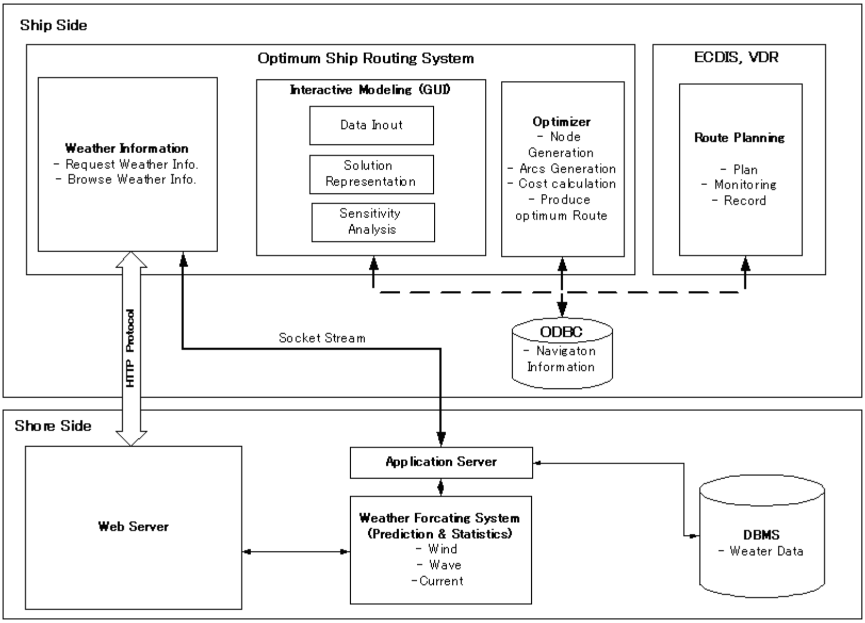


Fig. 7. Configuration of web-based distributed optimum routing system

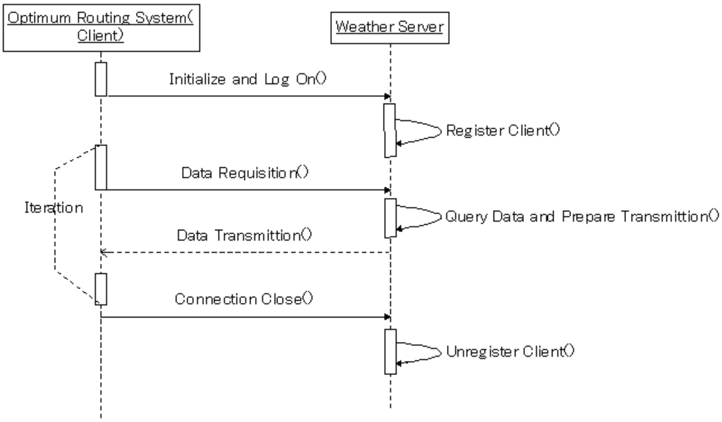


Fig. 8. Sequence diagram of socket connecting

5.2 Weather Server

It is clear that achieving the exact weather information plays a key role to produce more precise optimum route. To implement weather prediction facility into WOSRS is beyond the scope of this paper, so we limited the function of weather server only to send weather information stored in DBMS to clients using HTTP protocol. The configuration of Weather Server is shown at Fig. 10.

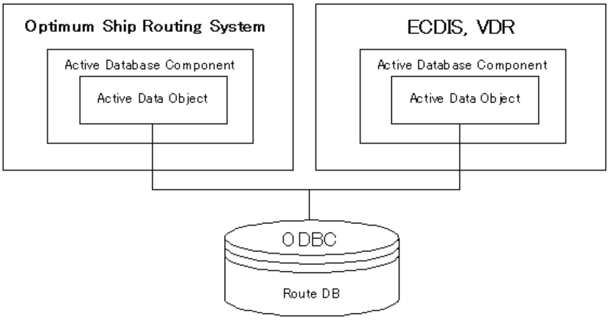

Optimum Ship Routing System
Active Database Component
Active Data Object
ECDIS, VDR
Active Database Component
Active Data Object
ODBC
Route DB

Fig. 9. Data sharing through ODBCS

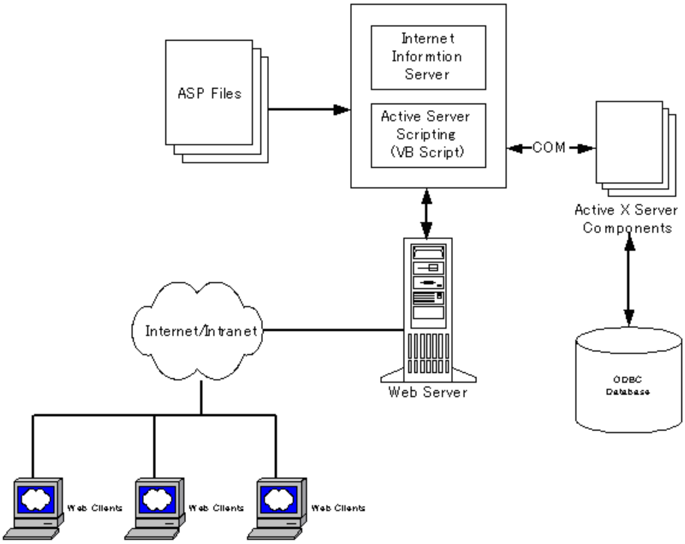

ASP Files
Internet Informtion Server
Active Server Scripting (VB Script)
COM
Active X Server Components
ODBC Database
Web Server
Internet/Intranet
Web Clients
Web Clients
Web Clients

Fig. 10. Configuration of weather server

5.3 Application Server

Application Server sends weather cell data to the connected clients via socket connection. While Weather Server handles the functionality concerning DBMS, The Application Server handles Security and User handling functionalities. Application Server keeps and manages client information and sends weather cell data according to clients' request. The message protocol is shown at Fig. 11. On receiving TTxProtol, the application severs carries out wave and current information to clients using TRxProtocol

In Fig. 12, you can see weather cell information distinguished by color, transmitted from application server. The spreadsheet on right corner shows weather cell information such as cell name, wave height, wave direction, current.

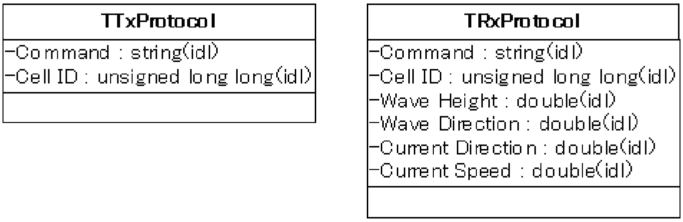
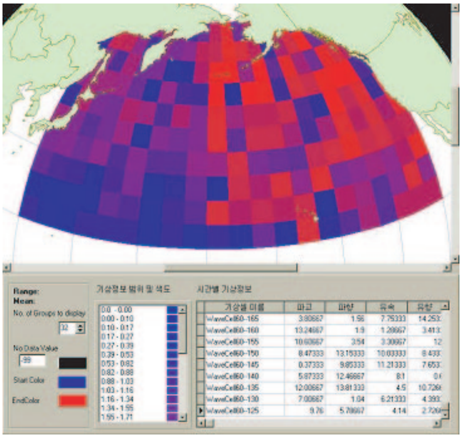

TTxProtocol
-Command : string(idl)
-Cell ID : unsigned long long(idl)
TRxProtocol
-Command : string(idl)
-Cell ID : unsigned long long(idl)
-Wave Height : double(idl)
-Wave Direction : double(idl)
-Current Direction : double(idl)
-Current Speed : double(idl)

Fig. 11. Message protocol



기상정보 범위 및 색도

0.0 - 0.00	🟦
0.00 - 0.10	🟦
0.10 - 0.17	🟦
0.17 - 0.27	🟦
0.27 - 0.39	🟦
0.39 - 0.53	🟦
0.53 - 0.82	🟦
0.82 - 0.88	🟪
0.88 - 1.03	🟪
1.03 - 1.16	🟪
1.16 - 1.34	🟪
1.34 - 1.55	🟪
1.55 - 1.71	🟪

시간별 기상정보

기상셀 이름	파고	파향	유속	유향
WaveCell60-165	3.80667	1.56	7.75333	14.253
WaveCell60-160	13.24667	1.9	1.20667	3.413
WaveCell60-155	10.60667	3.54	3.30667	12
WaveCell60-150	8.47333	13.15333	10.03333	8.433
WaveCell60-145	0.37333	9.85333	11.21333	7.653
WaveCell60-140	5.87333	12.46667	8.1	0.(
WaveCell60-135	12.00667	13.81333	4.5	10.726
WaveCell60-130	7.00667	1.04	6.21333	4.393
WaveCell60-125	9.76	5.70667	4.14	2.726

Fig. 12. Weather cell on client system

6 Conclusion

In this paper, previous literatures are surveyed and the results are summarized. A ship routing problem is formulated as a network model and is solved by a modified depth-first search algorithm. Because the authors approached to the ship routing problem as a navigator's point of view, more practical route can be produced than those of previous research. And WOSRS as subsystem of intelligent navigation system was designed to be a web based distributed system and implemented using ASP (Active Server Page) and ODBC.

The strong point of a network model is:

1. Isochrone Method and Dynamic programming methods can not produce a perfect Great Circle route even though there exists no interference on the route, but a network model based on great circle navigation methods can produce a perfect great circle route.
2. Isochrone Method produce only one optimum route, but network model can examine several alternative routes.

But this paper did not apply the real aspect of cost estimation in weather forecasting, so still further research required to use this model in practical navigation.

References

1. Hanssen, G.L., James, R.W.: Optimum Ship Routing. The Institute of Navigation, **13** (1960)
2. Chen, H.: A Stochastic Dynamic Program for minimum Cost Ship Route. Ph.D Thesis, Department of Ocean Engineering. Massachusetts Institute of Technology (1978)
3. Hagiwara, H.: Weather Routing of sail Assisted Motor vessels. Ph.D Thesis, Delft University, Nov (1989)
4. Papadakis, N.A.: On the Minimal Time Vessel Routing Problem. Ph. D. Thesis, Department of Naval Architecture and Marine Engineering. The Uni. of Michigan, Ann Arbor, Mich. (1988)
5. Perakis, A.N., Papadakis, N.A.: New Models for minimal time Ship Weather Routing. SNAME Transaction, **96** (1988)
6. Perakis, A.N., Papadakis, N.A.: Deterministic minimal time vessel routin. Operations research, **38** (1990) 426–438

Development of a Residential Gateway and a Service Server for Home Automation*

Jongkyu Park[1], Ilseok Han[1], Jinhyuck Kwon[1], Jun Hwang[1], Hagbae Kim[1],
Sangtae Ahn[2], and Whie Chang[2]

[1] Department of Electrical and Electronic Engineering,
Yonsei University, Seoul, Korea
hbkim@yonsei.ac.kr
[2] C-EISA Co., Ltd., Seoul, Korea

Abstract. We have developed a residential gateway that centralizes device interfacing between the external Internet and internal devices as well as appliance networks. We have also proposed the concept of a service server to be implemented for a large scale of apartment complex or a wide area house. The primary components of the residential gateway to be implemented in this paper include a processor(Motorola MPC8240), persistent storage(flash RAM, extend storage device), networking modules (such as TCP/IP for Ethernet, ADSL), home networking(HomePNA, IEEE1394, PLC), device interfaces(serial or PCI), home automation, and telecommunication system(PSTN/SLT, VoIP, Video Communication), which are typically powered by a certain RTOS. Finally, we have test results validating the effectiveness of both the residential gateway and the service server.

1 Introduction

As the information and communication technologies including computers, softwares, and the network are rapidly developed, the computer–related activities are getting more generalized even at home. As the broadband network infrastructure such as an xDSL is popularized, internet is also used frequently for a variety of purposes. Under the present conditions, people begin to use more than one computer connected with one another, and several home appliances are going to be related closely with the network. Therefore, the home network utilized for a simple terminal of the global network in the past is being expanded to another part of the sub network. Because it is relatively hard to establish additional network lines in the home area, the pre–constructed telephone line or the power cable can be effectively utilized as a proper transmission medium. In addition, the contemporary and new-future houses and apartments, the network lines

* This work is done by the MOCIE project; the Development of High-Performance Scalable Web Servers.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 137–149, 2002.

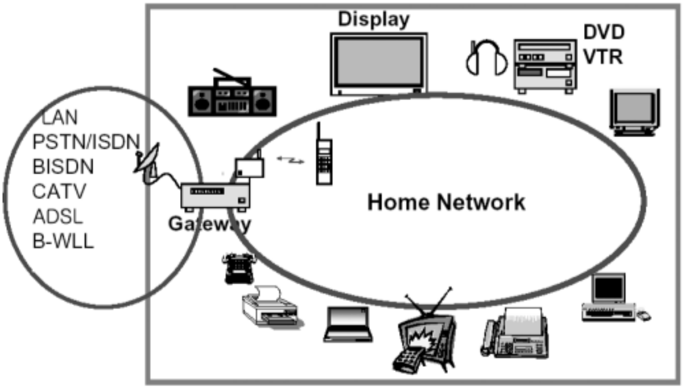


Fig. 1. Home Network

that are compatible with newly–proposed protocols need being constructed during the building processes. For a variety of connecting home–area protocols with the existing external network, we require a new gateway that does not only make the home-area network operating in the sub–network but also connects to the external network. A Residential Gateway(RG) interconnects home appliances by means of heterogeneous network paths such as power cables, telephone lines, RF and etc[1]. In the same way to the current gateways, the RG plays a key as a router that makes a subnet with connecting the entire home-area network, and translates different protocols to be matched to the xDSL or Ethernet established externally[2]. Furthermore, the RG becomes gradually to meet the requirements of the medium between Internet and the home appliances. It is necessary in the course of evolution from simple home appliances to Internet information appliances. (Note that a RG generally supports the OSGi standard as a midstage service transfer platform between the hardware and the system software.) In this sense, the home network can be constructed with a home server which locates in the center of Internet appliances, as depicted in Fig. 1. A home network in this paper does not mean the existing electrical wires but the new local connection system including the high-speed Internet communication and digital appliances. It differs from the conventional home automation schemes focused on low-speed simple data communications. However, there are a lot of problems to exchange existing network environments into new home networks. For example, IP address assigning that is a key to internet appliances can be a model problem. However, it is well known that the IP header has only 32bit address fields in IPv4. Thus, as upgrading home-network environments, home appliances need by for more IP addresses, which incurs a serious IP address shortage problem. Moreover, the adoption of dynamic IP address assigning in the current xDSL technology is an obstacle against RG's expansion to the home server. The dynamic method of assigning IP address to maintain is the IP address pool by sharing the limited IP addresses only when a certain network device requires. Therefore, the RG can be activated as a normal server with IP address confirmation when required to

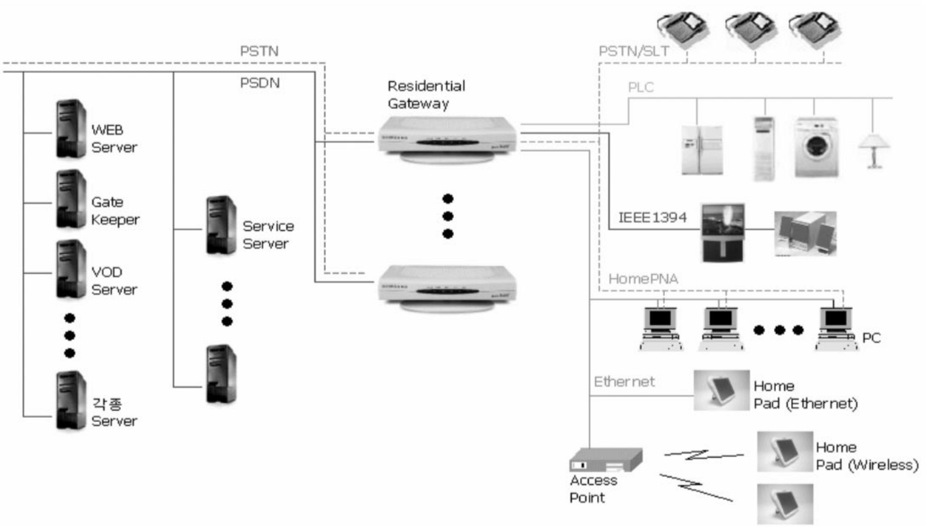


Fig. 2. Overview of the system

be connected to the external network. In this paper, we also develop and implement a Service Server(SS) that offers the integrated managements and control services for a variety of devices connected with the RG in the home area. The SS can monitor, manage, and control the devices that are connected through the wire and/or wireless methods to the external network for the internal RG as well as the home automation devices. The SS can not only solve the dynamic IP Address assigning problem but also assigns local IP addresses to the homenetwork devices through the Network Address Translation(NAT). It also provides somewhat useful functions for the home network and the RG for other additional services. In Section 2, we describe the basic structure of the RG by investigating a Real-Time Operating System(RTOS) for the home server, the software technologies of information home appliances including the multimedia middleware, and the information appliance applications. In Section 3, we explain the technologies associated with such key wire and/or wireless home networking as Phoneline Network Alliance(HomePNA), Power Line Communication(PLC), and IEEE1394. In Section 4, the structure and operation of the SS in cooperation with the RG are also described. The paper concludes with Section 5.

2 Home Gateway Architecture

As shown in Fig. 2, the basic structure of the RG to be implemented in this paper consists of such common servers as a service server, a gatekeeper, and a VOD server, which are equipped jointly at flexible places from a small house to a large apartment complex. The RG essentially enables the followings:

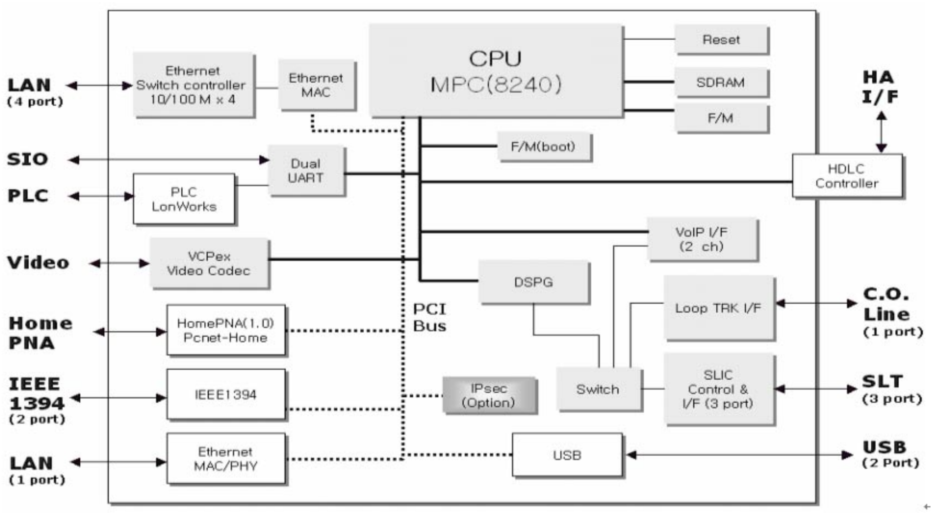


Fig. 3. Hardware architecture of the RG

- to offer more expanded services and capabilities to manage home appliances by using the SS,
- to self-upgrade the softwares of the RG and other appliances to correct or reconfigure those,
- to access to the home appliances from the external network even on an alias IP, and
- to offer substantial efficience and convenience for remotely managing and repairing the RG that needs not being in every housing complex or sub-network separated logically nor physically.

2.1 Hardware Architecture of the RG

The basic hardware architecture of the RG is well depicted in Fig. 3. A MPC8240 of Motorola CPU which has 350MIPs capacity with 200MHz internal clock. The RG requires high processing power to handle heavy workloads of tasks with the protocol stacks and the Java Virtual Machine management. The MPC8240 utilizes up to 32 pieces of PCI devices by a built-in PCI controller in the 32bit PowerPC core, and it has an effective memory mapping technique called PortX. The 4MB x 3 flash memory stores such fundamental data as the OS, the Java virtual machine, and the protocol stacks. After some initial processes, the data is loaded onto the 64MB memory region. The first 512 KB of flash memory region is used for the initial booting procedure. For Ethernet supporting, we separate the internal and the external access ports. A switching chip switches four ports for internal access. To support dual serial ports, one port is assigned to PLC of LonWorks while supporting two video ports. For reducing the CPU

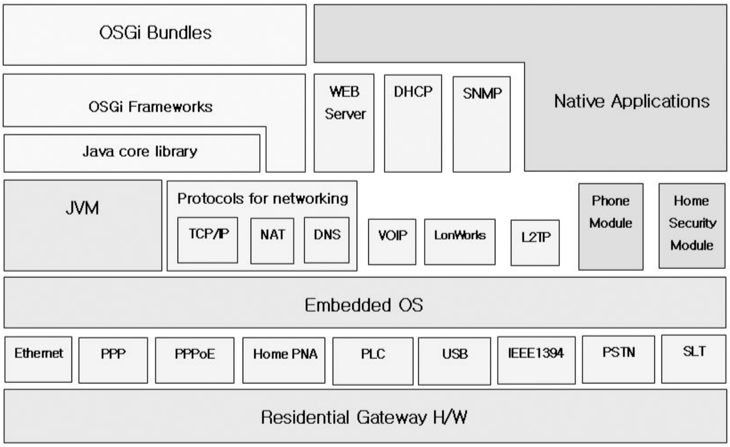


Fig. 4. Software architecture of the RG

load, the independent VCPex, DSPG, and IPsec chips operate data compression and dissolution processes.

2.2 Software Architecture of the RG

Our developed RG has a basic software structure as displayed in Fig. 4. All of the protocols and the applications are implemented on the OS as stack structures. This improves the manageability of the non-hardware part of the RG, which also helps immediate and concise software upgrading. The primary OS is based on a commercial RTOS, MQX2.40. Because the MQX offers all parts of kernel sources and various useful kernel functions, it is quite helpful for developing the specific OS and platforms. The device drivers are also implemented only for the requirements of the compatibility with the standard device driver interfaces of the MQX, as presented in Fig. 4. After modifying the parts of the OS and building the protocol stacks, we merge those with the application layer. However, because the OSGi bundle is not standardized yet, the RG temporarily satisfies in-company specification for the home appliances of a specific electronic company. (Here, we adopt those of Samsung Electronics.)

3 Home Network

3.1 Home Phoneline Networking Alliance (HomePNA)

The HomePNA is a network based on the phone lines constructed earlier at home, which allows one to equip the LAN at home without special equipments like a hub and a router as described in Fig. 5. The primary characteristics of the HomePNA is that additional building cost is low due to utilizing telephone lines already available and that one can still enjoy the existing telephone services.

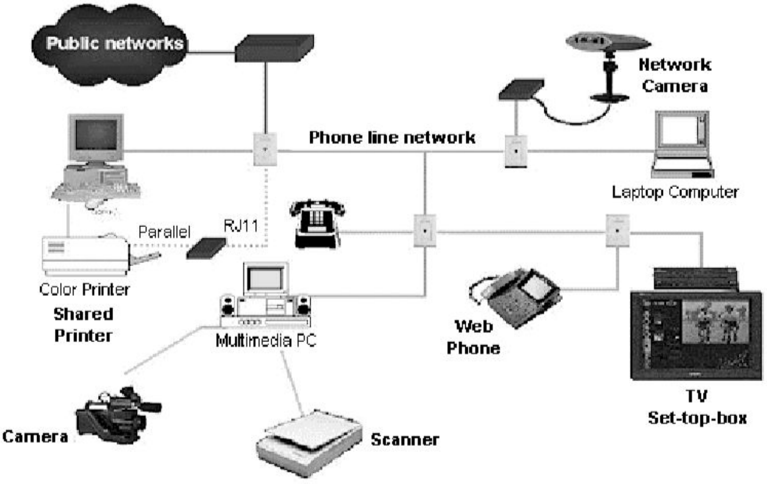


Fig. 5. Architecture of HomePNA

However, the RG should be considered for signal attenuations induced by high transmission losses of the telephone lines, and the random tree topology caused by several bridged tabs on the line. The network should be able to steadily operate under dynamic property changes of the transmission line conditions that could be created by telephone hookup or line failures.

3.2 IEEE1394

The RG also supports IEEE 1394, the high-speed serial bus for accepting the digital AV appliances. For the first time, Apple computer developed it by calling FireWire for replacing SCSI[3], and the society of IEEE chose it as the standard officially to call IEEE1394 which was basically developed as the interface among home appliances such as computer peripherals, video camera, audio device, television, videocassette, and VCR to readily be accessed to the home computer. It has three kinds of data transmission speeds or bandwidths, 100MB, 200MB, and 400MB, and supports the hot plug-in means appliances which can access directly to the computer and sixty three units can be accessed to it maximally. There are two basic data transmission methods in IEEE1394. One is isochronal transmission where as the other is asynchronous one. Because the former is real-time transmission, it is useful for transmitting the multimedia information like video or audio data. As the latter is partitioning the data, it is suitable for the transmission between the PC and the peripheral such as a hard disk or a printer. The applications using IEEE1394 are being continuously developed.

3.3 Power Line Communication (PLC)

Our RG also supports the PLC. Using the PLC, it is not necessary to construct the new lines for the home-area network. Several standards for the PLC are

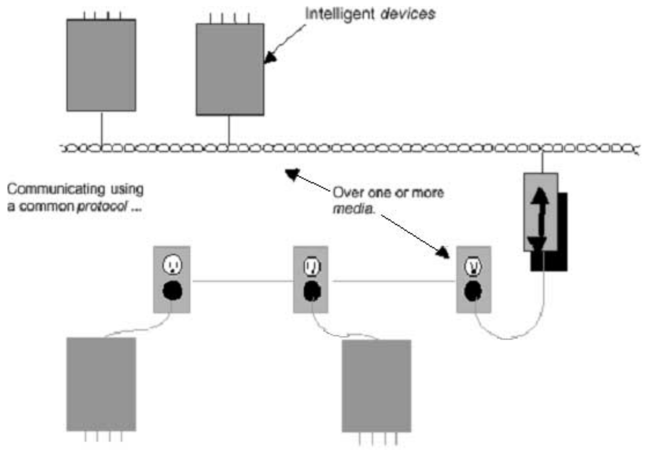


Fig. 6. Architecture of the PLC

proposed, however our RG picks up LonWorks that is a common form to be applied widely. LonWorks is the network control system that was originated from Echelon[4], which is operated on the protocol LonTalk. LonWorks is not a master-slave method but a peer-to-peer transmission method, that is, one node of LonWorks is in the equal status to another. In Fig. 6, the intelligent device is represented by each node, which consists of the neuron chip including the LonTalk protocol and the transceiver appropriate to the transmission media. Each neuron chip has an inherent 48-bit ID. The LonTalk can transmit data through AC or DC power lines as well as general twisted pair lines according to appropriate transceivers. As depicted in Fig. 6, the network of our system is organized with the twisted pair and the AC power line. We can communication through the LonWorks router among the networks on different physical media.

3.4 Ethernet

A contemporary apartment or a standalone house may include the communication line for the Ethernet inside the home with enhanced options. It should be able to connect among each node with the hub on the Ethernet. The RG supplies four ports of three-layer switching hubs. It provides one port which is separated from the hardware and supports Point-To-Point over Ethernet(PPPoE) for connecting with the ADSL external network. Our RG shares the IP's in terms of NAT. Because this structure is located on the bridge interconnecting inner and outer home network, all of the security problems should be solved through the RG[3].

3.5 Open Service Gateway Initiative (OSGi)

Our gateway follows OSGi standard as a middle stage service transfer platform between the physical system and the system software. The OSGi specification

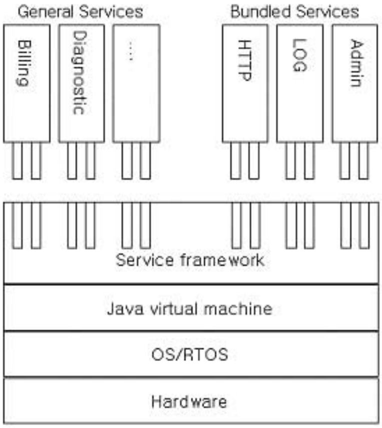


Fig. 7. OSGi

is a collection of standard API's that define a service gateway. They address service delivery and remote service administration as well as dependency, life cycle, resources and device management. The OSGi's core is a simply service framework. Its main goal is to use the Java language's platform independence and dynamic code–loading capabilities to simplify development and dynamic deployment of information-appliance applications to be implemented with physically poor resources. A service implies a self-contained component, accessible through a defined service interface that includes the Java classes to perform certain functions. An application is built around a set of cooperating services and can extend its functionality during the runtime by requesting more services. The framework registers and installs services, while maintaining a set of mappings to their implementations. It also manages the dependencies among those. A bundle denotes a functional and deploying unit for shipping services. While the framework manages service registration, unregistration, or property modification for the services and bundle's life cycles, it generates proper events according to each case. Fig. 7 shows how the services are accessed to the framework to be operated in [5]. This pluggable framework is ideal for just-in-time service delivery, because it allows the gateway device to download requested services over the network only when needed. The framework also supports service updating and versioning. It can be used not only for quickly checking the version of a service that is running at a gateway but also dynamically updating this service from a remote location. By using the scheme, the embedded devices checks its own status and warns the device manufacturer on any network abnormality using the home server.

3.6 VoIP

As depicted Fig. 8, the network between the gatekeeper and the gateway is composed of Public Switched Data Network(PSDN) which is a ISDN, xDSL or LAN,

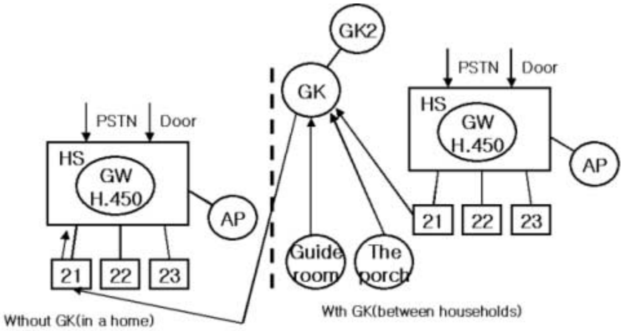


Fig. 8. scenario of VoIP communication

and Public Switched Telephone Network(PSTN). The gatekeeper is connected with PSTN as the external network. The terminals, gateways, and gatekeepers are implemented to control data stream, based on recommendation of H.323 protocol stacks. In this paper, the RG supplies required functions relevant to VoIP as follows; (i)the communication among terminals of other gateways in the same gatekeeper, (ii)Internet telephony, (iii)interphone functions among terminals in the same gateway, and (iv)general telephone calls. In the first case, the terminal(like PAD or PC) at one household calls the terminal at another household, which is located in the coverage of the same gatekeeper. The gatekeeper receives the call signal and transmits the signal to the gateway at the receiving household by using the table that maintaines alias IP database of each household. In the third case, each terminal communicates by using the gatekeeper's alias IP table. The PSDN and the PSTN are used for the second case and the fourth, respectively.

The major operation of the the RG associated with VoIP is that the gatekeeper stores the alias IP's of relevant terminals and calls the terminal according to the alias IP table. The gateway transforms VoIP into analogue telephone signals, or vice versa. Fig. 8 represents this VoIP communication scenario.

3.7 Video Communication

The H.323 protocol stack supports voice communication by default, and video and data communication optionally. The gateways, terminals, and the gatekeepers in this paper are capable of supporting all of voice, video and data communications. After the voice and video data are created through the video camera, the gateway receives, packetizes, and synchronizes the data to exchange those with each terminal through video communication. Because the transmitting terminals and the receiving terminals communicate with each other in packet types, both video and data communication mechanisms are quite similar to VoIP.

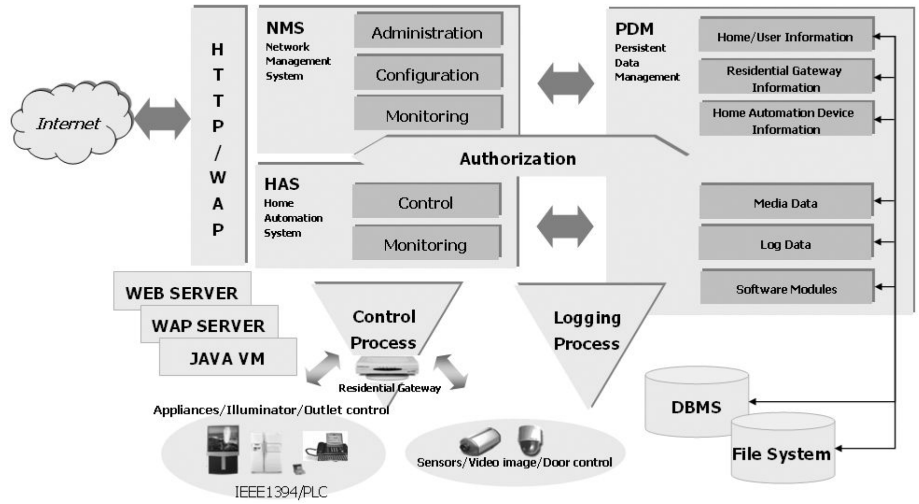


Fig. 9. System architecture of the SS

3.8 Public Switched Telephone Network (PSTN)/ Single Line Telephone (SLT)

The RG operates like the switch for a common telephone network at home. The RG provides one port of the public line (PSTN) and three ports of the private line (SLT), and switches the latter three ports for each line to share the public line. These private lines can be connected with the Internet telephony service through VoIP. The RG supports the communication functions among each home-area terminal such as the telephone or the home pad through the private line.

4 SS System Architecture

The SS is a server to offer the integrated managements and with control high-level services for various kinds of devices connected with the RG. It can also monitor the devices those are connected through the wire or wireless channel to the external network. Fig. 9 shows the basic system architecture of the SS, which is constructed by the function of Authorization, Network Management System(NMS), Home Automation System(HAS), and Management. All functions associated with the management, monitoring, and control can be executed on the web browser, through the security enhanced HTTPS protocol[1][2][6].

4.1 Authorization

There are two user levels; an administrator and a general user. The administrator has an authority to monitor configurating the SS as well as investigating the

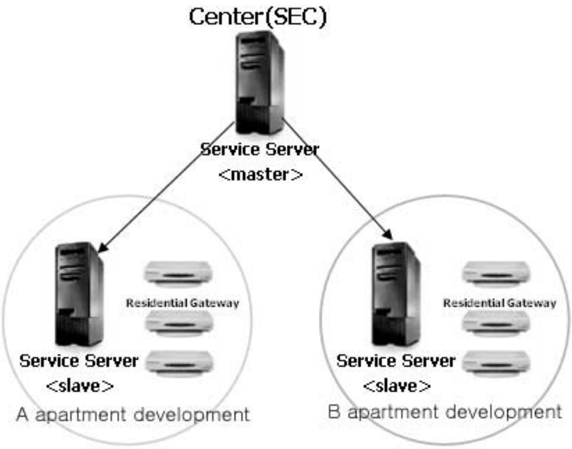


Fig. 10. Master-Slave structure

status of the RG and the HA devices installed in the entire domain. The administrator establishes the access authorization for the RG and the HA devices for general users, who have limited different classes of authorization for managing and monitoring for both of the RG and the HA devices. A certain general user can reestablish the access authorization level by the individual. The SS offers an Access Control Table(ACL) for providing proper information about each user.

4.2 Network Management System (NMS)

The NMS of the SS manages the running software and configures the HA devices and all registered RG's. The administrator can also establish the ACL for handling user registrations, service programs, and network port configuration. The administrator monitors the SS and all devices those are registered in the specific domain. Moreover, various statistical data stored in DB can be reported by simple interfaces such as tables or graphs. The SS manages and controls in the center by the top-down method on the basis of Master-Slave structure. As shown in Fig. 10, the master SS in the center manages and controls the slave SSs deployed in the apartments A and B.

4.3 Home Automation System (HAS)

The HAS controls each individual device connected with the RG. It provides a user-friendly GUI which is based on the information acquired by automatic or manual configuration awareness method. Because different devices generally require different control mechanisms, the control function usage of the GUI is statically provided. However, by using the OSGi Framework, the HAS dynamically downloads the library(bundle) which can control specific devices. The HAS provides various GUI's to control the HA devices, for assigning the unique controller types such as a PC, a home PAD, a PDA, and a cellular phone. The SS offers both HTTP and WAP protocols if the device has remote accessability. [1][7].

4.4 Persistent Data Management (PDM)

The PDM stores and manages for the persistent data generated by the HA devices and all registered RGs in the specific domain. The persistent data contains all information associated with the home and the users, the RG, home automation devices, media data, and log messages.

4.5 Software Management

The Software Management checks the authorizations required by the RG for the version checking and upgradeing of the software. In contrary, the RG avoids hacking by authenticating the SS. The RG has the flexible structure which takes various versions of the software in each module for efficient upgrade. The effective management of the version history makes it easy to rollback to the previous version if required.

4.6 Security Management

The Security management is divided into Intrusion Detection System(IDS) and data security. By using the IDS the SS improves the security level, which offers the functions of a two-level protection mechanism. It can detect and block the intrusion of unauthorized data access. A firewall service is required for the first level of security guarantee. Building the IDS according to the demanded security level, it detects in advance illegal accesses to the RG. Approaches through the web can be blocked by the HTTPS, which encapsulates the data before transmission. In case of TELNET, both login ID and password can be encrypted by loading the Secure Shell(SSH) access.

5 Conclusion

The implemented RG in this paper has somewhat distinguished characteristics as follows. It does not only guarantee the high-speed data transmission among home automation devices placed in the home but also has the standards of common access methods connecting the communication devices and home appliances through both external and internal networks. It maximizes the degree of utilization of currently–available telephone lines and CATV coaxial cables while minimizing the additional cost for newline installation. The RG can be embedded in an independent topology with other access networks, which makes it possible to achieve independent network topologies at single residence. The high adoptability of the RG helps the unification of heterogeneous communication lines at home. It also prevents the overlapping investments for infrastructure and provides the adoptability to the technologies for subscriber services, for example, combining broadcasting and tele/internet communication, offering multimedia services, using wire and/or wireless channels. The RG is, thus, basically suitable for a large scale of the apartment complex employing the SS, to be available for new services by integrating a number of functions for the home server. It offers an interface with the HA devices which is evolving in various ways. The home gateway keeps developing with the changes of external networks, home networks, access networks, and market trends. To meet the rapidly-changing customer's requirements, the existing plug-in type home networking modules will gradually be changed into the form of integrated box. Therefore, it is urgent to make a standard for the core devices of the home gateway which can support various multimedia services, deliveries, and shares the data provided by the network devices at home through Internet.

References

1. T. Saito, I. Tomoda, Y. takabatake, J. Ami, K. teramoto. :Home Gateway Architecture and Its Implementation,IEEE Trans on Consumer Electronics. **46** (Nov 2000) 1161–1165
2. Cahners In-Stat Group : Residential Gateways:The Heart of the Home Network, Report No.LN9909HN, 5–12
3. K. Shimada, H. Sasaki, Y. Noguchi. : The home networking system based on IEEE1394 and Ethernet technologies, (2001). ICCE. International Conference on Consumer Electronics, (2001) 234–235
4. Echelon Co,. : Introduction to the LonWorks System, Echelon Co. general manual part No.078-0183-01A
5. Li Gong : A software Architecture for Open Service Gateways, IEEE Internet Computing, (Jan.-Feb. 2001) 66–70
6. H. Desbonner, P.M. Corcoran : System Architecture and Implementation of a CEBus/Internet Gateway, IEEE Trans on Consumer Electronics. **43** (Nov 1997)
7. M. Hashimoto et al. : A Home Network Architecture Considering Digital Home Appliances, IWNA98

Performance of Multiple TCM Codes with Transmit Antenna Diversity

Uiyoung Pak[1], Joonseok Maeng[2], Iksoo Jin[3], and Keumchan Whang[2]

[1] Agency for Defense Developement, Daejeon, Korea
[2] Department of Electrical and Electronic Engineering,
Yonsei University, Seoul, Korea
kcwhang@yonsei.ac.kr
[3] Div. of Information and Communication Engineering,
Kyungnam University, Masan, Korea

Abstract. Space-time block codes (STBC) have no coding gain but they provide a full diversity gain with relatively low encoder/decoder complexity. Therefore, STBC should be concatenated with an outer code which provides an additional coding gain. In this paper, we consider the concatenation of multiple trellis-coded modulation (MTCM) codes with STBC for achieving significant coding gain with full antenna diversity. Using criteria of equal transmit power, spectral efficiency and the number of trellis states, the performance of concatenated scheme is compared to that of previously known space-time trellis codes (STTC) in terms of frame error rate (FER). Simulation results show that MTCM codes concatenated with STBC offer better performance on slow fading channels, especially much better on fast fading channels, than previously known STTC with two transmit antennas and one receive antenna.

1 Introduction

Space-time block codes (STBC) are one of remarkable modulation techniques, which were first discovered in [1]. Using 2 transmit antennas and M receive antennas, STBC provides a diversity order of 2M at the receiver. STBC does not require any feedback from the receiver to the transmitter. An important issue is the ability of STBC to obtain maximum advantage under different fading situations. STBC is not designed to achieve an additional coding gain. Therefore, STBC should be concatenated with an outer code which provides an additional coding gain. In [2], it was shown that optimal trellis codes designed for AWGN channels are also optimal for use with STBC in slow Rayleigh fading environments in terms of error event probability. But the results are not necessarily optimal if the channel conditions are changed, or if bit error rate (BER) or frame error rate (FER) is considered as a performance measure.

Divsalar et al. [3] have shown that the multiple trellis-coded modulation (MTCM) codes, transmitting k^* M-ary signals per trellis branch, provide performance gains on the fading channel. Periyalwar et al. [4] have modified the MTCM codes and proved that additional gain can be obtained on Rician fading channels. In fading channels, MTCM codes show better performance than

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 151–159, 2002.

Ungerboecks TCM codes [8] for the increased design parameters and additional temporal diversity effect when fading variation is relatively fast. There are several papers on the concatenation of TCM codes with STBC [5], but very few on the concatenation of MTCM codes with STBC. The main purpose of this paper is to investigate and validate the use of MTCM codes as an outer code with STBC.

In this paper, we consider the concatenation of MTCM codes with STBC for achieving additional coding gain with full antenna diversity. Furthermore, a closed form of the pairwise error probability is obtained for both fast and slow fading channel under the assumption of perfect channel state information. Using criteria of equal transmit power, spectral efficiency and the number of trellis states, the performance of concatenated scheme is compared to that of previously known space-time trellis codes (STTC) in terms of FER.

2 System Model

We consider a wireless space-time coded systems with two transmit antennas and one receive antenna. The channel is frequency non-selective fading and it is assumed that channel state information is known to receiver perfectly. The received signal at time t is given by

$$r_t = \sum_{i=1}^{2} h_t^i c_t^i \sqrt{E_s} + \eta_t \tag{1}$$

where h_t^i is the complex Gaussian channel path gain from transmit antenna i; E_s is the energy per symbol; c_t^i is space-time block coded symbol transmitted via transmit antenna i at time t ; η_t is the additive complex white Gaussian noise at time t with zero mean and variance $N_0/2$ per dimension.

The amplitude of the envelope of the received signal is a normalized random variable with a Rician probability density function given by

$$p(\left|h_t^i\right|) = 2\left|h_t^i\right|(1+K)e^{-K-\left|h_t^i\right|^2(1+K)} I_0(2\left|h_t^i\right|\sqrt{K(1+K)}), \qquad \left|h_t^i\right| \geq 0 \tag{2}$$

where the Rician fading parameter K represents the ratio of the direct and specular signal components to the diffuse component and $I_0(\cdot)$ is the zero order modified Bessel function of the first kind. As a special case, $K = 0$ yields Rayleigh fading, and $K = \infty$ describes the AWGN channel.

Fig. 1 illustrates the block diagram of the considered system employing MTCM concatenated with STBC. In Fig. 1, the block size of the interleaver is N in symbol duration and the interleaver output is defined by mapping function $f(t)$. STBC encoder/decoder blocks are used for giving full antenna diversity gain to conventional MTCM codes and they consist of relatively simple block coding technique. Then, the outputs of the STBC decoder are sent to MTCM decoder using the conventional Viterbi algorithm.

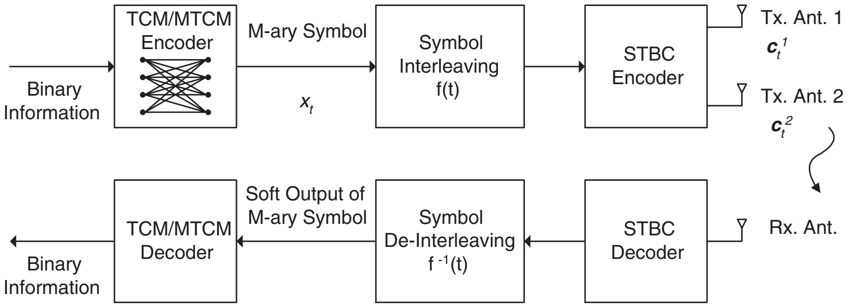


Fig. 1. Concatenated system of TCM/MTCM codes with STBC

3 Performance Analysis

Assuming that the MTCM encoder sends a codeword $\mathbf{x} = (x_1 x_2 \cdots x_t \cdots x_l)$ to the STBC encoder where l is the codeword length and x_t is a trellis-coded symbol at time t. Then, space-time block coded symbol $\mathbf{c}_t = (c_t^1, c_t^2)$ is transmitted to receiver via fading channel and it is assumed that the fading coefficient h_t^i is constant across two consecutive symbols for STBC decoding [1], i.e., $h_{2n-1}^i = h_{2n}^i$.

It is assumed that a codeword $\mathbf{x}$ was transmitted, the conditional pairwise error probability $p\left(\mathbf{x} \to \mathbf{x}' \left| \mathbf{h}^1, \mathbf{h}^2 \right.\right)$ that a maximum likelihood decoder erroneously chooses the codeword $\mathbf{x}' = (x_1' x_2' \cdots x_t' \cdots x_l')$ can be written as

$$\begin{aligned} p\left(\mathbf{x} \to \mathbf{x}' \left| \mathbf{h}^1, \mathbf{h}^2 \right.\right) &\le \Pr\left\{ m\left(\mathbf{r}, \mathbf{x}'; \mathbf{h}^1, \mathbf{h}^2\right) \ge m\left(\mathbf{r}, \mathbf{x}; \mathbf{h}^1, \mathbf{h}^2\right) \right\} \\ &= \Pr\left\{ \sum_n m\left(r_{2n-1}, r_{2n}, x'_{f(2n-1)}, x'_{f(2n)}; h_{2n-1}^1, h_{2n-1}^2, h_{2n}^1, h_{2n}^2 \right) \right. \\ &\qquad \left. \ge \sum_n m\left(r_{2n-1}, r_{2n}, x_{f(2n-1)}, x_{f(2n)}; h_{2n-1}^1, h_{2n-1}^2, h_{2n}^1, h_{2n}^2 \right) \right\} . \end{aligned} \quad (3)$$

Decoding error occurs when soft decision metric $m\left(\mathbf{r}, \mathbf{x}'; \mathbf{h}^1, \mathbf{h}^2\right)$ is larger than metric $m\left(\mathbf{r}, \mathbf{x}; \mathbf{h}^1, \mathbf{h}^2\right)$, where $\mathbf{r}$ and $\mathbf{h}^i$ are given by $\mathbf{r} = (r_1 r_2 \cdots r_t \cdots r_l)$, $\mathbf{h}^i = \left(h_1^i h_2^i \cdots h_t^i \cdots h_N^i\right)$, respectively.

After some manipulation by using the Chernoff bound, the pairwise error probability conditioned on the fading coefficient $\mathbf{h}^1, \mathbf{h}^2$ is well approximated by

$$\begin{aligned} p\left(\mathbf{x} \to \mathbf{x}' \left| \mathbf{h}^1, \mathbf{h}^2 \right.\right) \le \exp\left\{ -\frac{E_s}{4N_0} \sum_n \left(\left|h_{2n}^1\right|^2 + \left|h_{2n}^2\right|^2 \right) \right. \\ \left. \times \left(\left| x_{f(2n-1)} - x'_{f(2n-1)} \right|^2 + \left| x_{f(2n)} - x'_{f(2n)} \right|^2 \right) \right\} . \end{aligned} \quad (4)$$

We may rewrite (4) as

$$p\left(\mathbf{x} \to \mathbf{x}' \left| \mathbf{h}^1, \mathbf{h}^2 \right.\right) \le \exp\left(-\frac{E_s}{4N_0}\sum_{t\in\eta}\left(\left|h^1_{f^{-1}(t)}\right|^2 + \left|h^2_{f^{-1}(t)}\right|^2\right)\left|x_t - x'_t\right|^2\right) \quad (5)$$

where η is the set of all t such that $x_t\ /\!\!\!= x'_t$.

3.1 Fast Fading Channels

Let us assume that codeword lengths tend to be much smaller than the interleaver size, i.e., $N \gg L$. In that case, the entire $\left|h^1_{f^{-1}(t)}\right|, \left|h^2_{f^{-1}(t)}\right|$ in $t \in \eta$ are independent samples of Rician random variables with Rician parameter K in fast fading channels. Then the pairwise error probability can be obtained by averaging (5) over the pdf of $\mathbf{h}^1, \mathbf{h}^2$ and can be written as

$$p(\mathbf{x} \to \mathbf{x}') \le \prod_{t\in\eta}\left(\frac{1+K}{1+K+\frac{E_s}{4N_0}\left|x_t - x'_t\right|^2}\exp\left(-\frac{K\frac{E_s}{4N_0}\left|x_t - x'_t\right|^2}{1+K+\frac{E_s}{4N_0}\left|x_t - x'_t\right|^2}\right)\right)^2. \quad (6)$$

If $K \to 0$, the channel yields the Rayleigh fading channel and (6) can be written as

$$p(\mathbf{x} \to \mathbf{x}') \le \prod_{t\in\eta}\left(\frac{E_s}{4N_0}\left|x_t - x'_t\right|^2\right)^{-2}. \quad (7)$$

If $K \to \infty$, the channel describes the AWGN channel, and (6) can be approximated to (8)

$$p(\mathbf{x} \to \mathbf{x}') \le \exp\left(-\frac{E_s}{4N_0}\sum_{t\in\eta}\left|x_t - x'_t\right|^2\right)^2 = \exp\left(-\frac{E_s}{4N_0}\left\|\mathbf{x} - \mathbf{x}'\right\|^2\right)^2. \quad (8)$$

It is worthwhile to note that in (7), the pairwise error probability of the MTCM codes concatenated with STBC varies inversely proportional to the product of the squared Euclidean distances to the second in Rayleigh fading channel (i.e., $K \to 0$). As the Rician parameter K approaches to infinity, the pairwise error probability largely depends on the squared Euclidean distances between the correct and the incorrect codeword. When K is moderate value (i.e., $0 < K < \infty$), the code performance can be explained as follows. At the low E_s/N_0, the pairwise error probability is greatly influenced by the squared Euclidean distances between the correct and the incorrect codeword. It is important to note that the product of the squared Euclidean distances can be used as a dominating parameter for the pairwise error probability at high E_s/N_0.

3.2 Slow Fading Channels

Now it is assumed that the entire fading coefficient in error event path is constant in slow fading environment, i.e., $h^i_t = h^i$ for all t. Then the pairwise error probability can be obtained by averaging (5) over the pdf of h^1, h^2 and can be written as

$$p(\mathbf{x} \to \mathbf{x}') \leq \left(\frac{1+K}{1+K+\frac{E_s}{4N_0}\sum\limits_{t\in\eta}\left|x_t - x'_t\right|^2} \times \exp\left(-\frac{K\frac{E_s}{4N_0}\sum\limits_{t\in\eta}\left|x_t - x'_t\right|^2}{1+K+\frac{E_s}{4N_0}\sum\limits_{t\in\eta}\left|x_t - x'_t\right|^2} \right) \right)^2 . \quad (9)$$

If $K \to 0$ (Rayleigh fading), (9) can be written as

$$p\left(\mathbf{x} \to \mathbf{x}'\right) \leq \left(1 + \frac{E_s}{4N_0}\sum_{t\in\eta}\left|x_t - x'_t\right|^2\right)^{-2} \leq \left(\frac{E_s}{4N_0}\left\|\mathbf{x} - \mathbf{x}'\right\|^2\right)^{-2} . \quad (10)$$

If $K \to \infty$ (AWGN), (9) can be approximated to (8). In slow fading channels, the pairwise error probability of MTCM concatenated with STBC varies on the squared Euclidean distances between the correct and the incorrect codeword for any Rician parameter K, and it is also expected that code with large free distance will show better performances.

4 Simulation Results

In this section Monte Carlo simulations of the considered codes are achieved. We consider transmitting a frame of 192 information bits in 20 msec. The 12×8 block interleaver is used for symbol interleaving. The carrier frequency is 2 GHz and it is assumed that the channel state information is known to the receiver. For outer code of STBC, we consider TCM and MTCM codes with 4-state and 8-PSK signal constellation proposed by Ungerboeck [8], Divsalar [3], and Periyalwar [4] for spectral efficiency of 2 bit/s/Hz, which have the same number of state and spectral efficiency as the STTCs proposed by Tarokh [6] and Firmanto [7]. In Fig. 3 - Fig. 5, the FER curves are plotted against the signal-to-noise ratio, defined as $SNR = 2E_s/N_0$.

Fig. 3 shows the FER of the considered space-time codes on fast Rayleigh fading channels. It is worthwhile to mention that diversity advantage refers to the slope of the FER curves versus SNR. In Fig. 3, it is shown that the Divsalar code with STBC and the Periyalwar code with STBC have steeper slope than the Ungerboeck code with STBC and the STTCs. At the FER of 10^{-2}, the performance gain of the Divsalar code with STBC over previously known STTCs is about 7dB. This is due to the fact that the Divsalar code with STBC and the Periyalwar code with STBC can achieve additional temporal diversity gain using multiple modulation symbols per trellis branch, while the STTCs and the Ungerboeck code with STBC have the diversity order of 2. It is well known that some space-time trellis codes achieve an additional horizontal shift of the FER

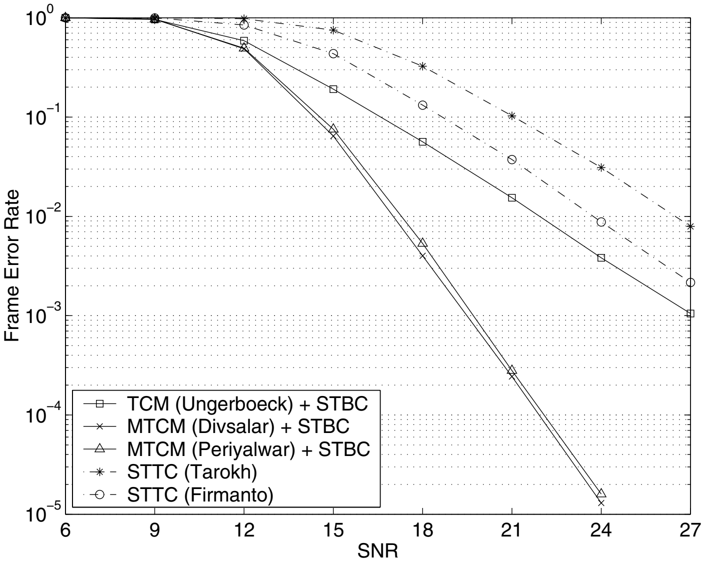


Fig. 2. Performance of space-time codes on fast Rician fading channel ($K = 10$ dB, 2 bits/s/Hz, mobile speed =120 km/hr)

curves which is called coding gain. At the FER of 10^{-2}, the coding gain of the Ungerboeck code with STBC over the Tarokh code is about 4 dB, and is about 2dB over the Firmanto code. It is in good agreement with [7] that the Firmanto code shows better performance than the Tarokh code, but the concatenated codes outperform the two kinds of the STTCs (i.e., Tarokh code and Firmanto code).

Fig. 2 illustrates the performance of the considered space-time codes on fast Rician fading channel with Rician parameter $K = 10$ dB. It is noted that as predicted by the performance analysis results in the previous section, the Ungerboeck code with STBC shows better performance than any other scheme. This result can be explained by the fact that the free distance of the Ungerboeck code with STBC is the largest among the considered TCM or MTCM codes. It is also noted that as the SNR increases, the Periyalwar code with STBC outperforms the other considered codes. At the FER of 10^{-2}, the Ungerboeck code with STBC offers an improvement of about 3.5dB over the Tarokh code on fast Rician fading channels. In Fig. 4 - Fig. 5, it is shown that the performance of the considered space-time codes on slow fading channels. In slow fading channel, as previously mentioned, the FER is mainly determined by the free distance, and it is worthwhile to notice that the Ungerboeck code with STBC shows the best performance among the considered codes. In slow Rayleigh fading channels, the coding gain of the Ungerboeck code with STBC over Tarokh code is about 3 dB at the FER of 10^{-2}, which was also observed in [5].

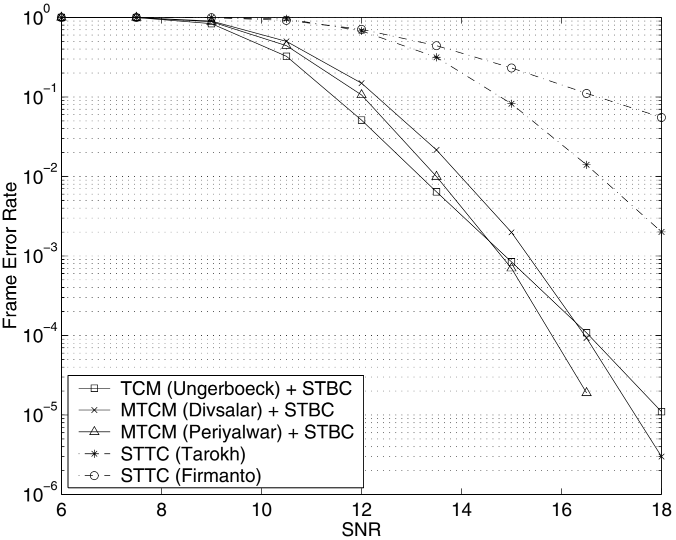


Fig. 3. Performance of space-time codes on fast Rayleigh fading channel (2 bits/s/Hz, mobile speed =120 km/hr)

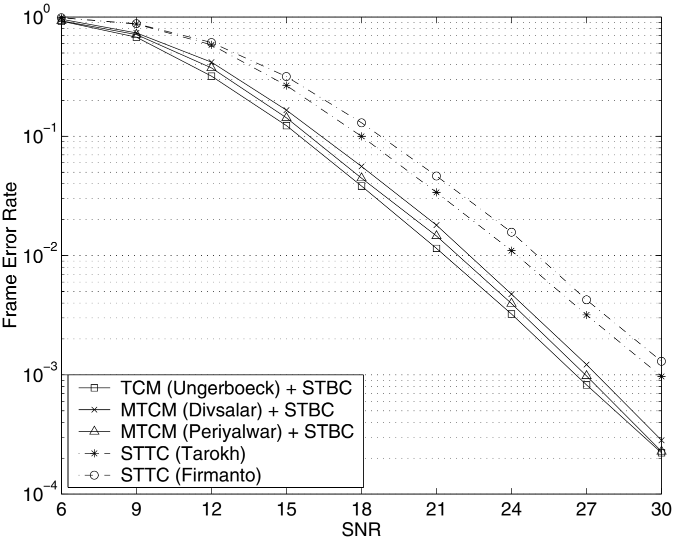


Fig. 4. Performance of space-time codes on slow Rayleigh fading channel (2 bits/s/Hz, mobile speed =3 km/hr)

In both fast and slow fading channels, the MTCM codes concatenated with STBC offers better performance than the previously known STTCs. Moreover, the MTCM concatenated with STBC shows much better performances than the TCM concatenated with STBC on fast fading channels, and almost the same

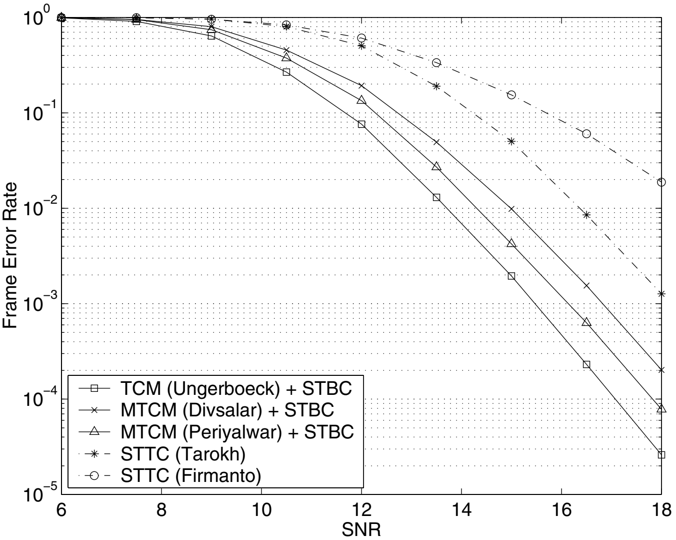


Fig. 5. Performance of space-time codes on slow Rician fading channel ($K = 10$ dB, 2 bits/s/Hz, mobile speed =3 km/hr)

performances as the TCM concatenated with STBC on slow fading channels. It is worthwhile to mention that the significant coding gain in addition to the diversity advantage can be achieved while the decoding complexity is mainly determined by the trellis complexity of the outer codes.

5 Conclusions

We have shown that the MTCM concatenated with STBC outperforms previously known STTCs with two transmit antennas and one receive antenna in both fast and slow fading channels. From the simulation results and the analytical comparison, it is shown that the performance improvement of MTCM concatenated with STBC is significant in fast fading channels and the improvements are mainly resulted from the additional temporal diversity gain and coding gain. In slow fading channels, the performance of MTCM concatenated with STBC is also better than the previously known STTCs. Moreover, it is shown that the performance improvement can be obtained while keeping the complexity almost the same as the STTCs.

References

1. Alamouti, S.M.: A Simple Transmit Diversity Technique for Wireless Communications. IEEE J. Select. Areas Commun. **16** (1998) 1451-1458
2. Alamouti, S.M., Tarokh, V., and Poon, P.: Trellis-Coded Modulation and Transmit Diversity - Design Criteria and Performance Evaluation. in Proceeding of ICUPC (1998) 703-707

3. Divsalar, D. and Simon, M.K.: The design of trellis coded MPSK for fading channels - Set partitioning for optimum code design. IEEE Trans. Commun. **36** (1988) 1013-1021
4. Periyalwar, S.S. and Fleisher, S.M.: A modified design of trellis-coded MPSK for fading channel. IEEE Trans. Commun. **41** (1993) 874-882
5. Sandhu, S., Heath, R., and Paulraj, A.: Space-Time Block Codes versus Space-Time Trellis Codes. in Proceeding of ICC (2001) 1132-1136
6. Tarokh, V., Seshadri, N., and Calderbank, A.R.: Space-Time codes for high data rate wireless communication - Performance criteria and code construction. IEEE Trans. Inform. Theory **44** (1998) 744-765
7. Firmanto, W., Vucetic, B.S., and Yuan, J.: Space-Time TCM with Improved Peformance on Fast Fading Channels. IEEE Commun. Lett. **5** (2001) 154-156
8. Ungerboeck, G.: Channel coding with multilevel /phase signals. IEEE Trans. Inform. Theory **28** (1982) 55-67

On the Study of MAC Layer for cdma2000 Wireless Packet Data Service

Sangyoub Kim

Nortel Networks, Wireless Network Engineering
2221 Lakeside Blvd.
Richardson, Texas 75082, USA
sangyoub@norltelnetworks.com

Abstract. Operators require new features and capabilities for current wireless systems to support increasing demand of high speed data services. Packet Control Unit (PCU) is a new node in a Base Station Controller (BSC) to provide high speed packet data service. The major responsibilities of PCU are as follows: segmentation/assembly of Point-to-Point (PPP) frames to/from ACN stream, session management, flow control and congestion control, DCR PLICF, and short data burst management for dormant calls. In this paper, Medium Access Control (MAC) layer in the PCU is analyzed for cdma2000 wireless network. With respect to QoS functionality in MAC layer, session setup delay is observed. In the simulation, it is assumed that the maximum transmission rate for forward link is 153.6 kbps while the transmission rate for reverse link is fixed to 9.6 kbps.

1 Introduction

The cdma2000 standard is essentially a stepping stone to true 3G technologies and it enables to offer significant benefits and capabilities to both users and service providers. Most notable among these are high packet data for users and better usage of the RF link for service providers. It is also important to note that the cdma2000 can be implemented in exiting network infrastructures (IS-95) with little new hardware requirement. One of new hardware features of the cdma2000 is PCU which provides the BSC connectivity to the Packet Data Service Node (PDSN). In this paper, Medium Access Control (MAC) layer in the PCU is analyzed for cdma2000 wireless network. With respect to QoS functionality in MAC layer, session setup delay is observed. In the simulation, it is assumed that the maximum transmission rate for forward link is 153.6 kbps while the transmission rate for reverse link is fixed to 9.6 kbps.

A block diagram of wireless communication system is presented in the Fig. 1.

1.1 Base Station Transceiver System (BTS)

The BTS is the cell site equipment that links the mobile units with the Base Station Controller (BSC). The BTS incorporates the transmitter, receiver, power

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 161–167, 2002.

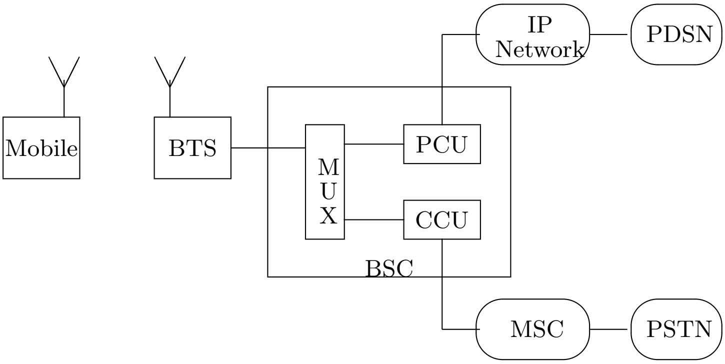


Fig. 1. A block diagram of wireless communication system

amplifiers, timing, and channel signal processing. Multiple BTSs are connected to the BSC via T1/E1 links. The bandwidth of T1 link is 1.536 Mbps and 2.048 Mbps bandwidth for E1 link. For cdma2000, the BTS has new features to handle not only voice traffic but also packet data traffic.

1.2 Base Station Controller (BSC)

The primary functions of BSC are voice coding, echo cancellation, processing handoff related messages, and providing power control. For IS-95, the BSC contains the resources, for example, Circuit Control Unit (CCU), for setting up and maintaining circuit voice and data traffic channels between BTS and Public Switched Telephone Network (PSTN). To support high speed packet data service for cdma2000, Packet Control Unit is introduced and is responsible for the termination point for the BSC connection to the PDSN. The physical connection between PCU and PDSN is either 10Base-T Ethernet (10 Mbps) connection or OC-3 (156 Mbps) connection to support high speed data throughput.

2 PCU Software

Fig. 2 illustrates the software entities of the PCU and detailed description of each entity can be found in [1]. The following subsections provide a high level view of the PCU software.

2.1 MAC Control Plane Entity

DCR PLICF: Dedicated Common Router Physical Layer Independent Control Function is responsible for providing the control and status information to the corresponding DCR data plane entity. It is also responsible for setting up and tearing down packet sessions. DCR PLICF operates in three states: *Null*, *Active*, and *Dormant*. The *Null* state is its default state prior to activation of

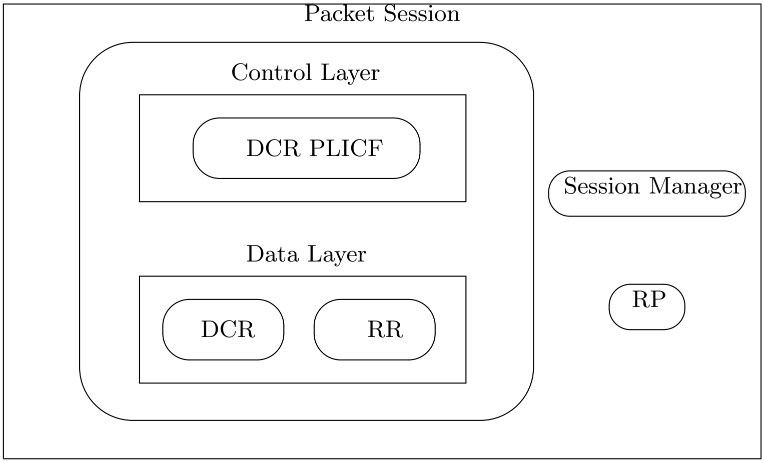


Fig. 2. Architecture of the PCU

data service. When the data service is invoked and after the service option is connected, then it transitions to the *Active* state. When the dedicated resources which were assigned to the packet data service are released, the DCR PLICF transitions to the *Dormant* state. In the *Dormant* state, the DCR PLICF directs packet data traffic through short data bursts over a common traffic channel.

Session Manager: The Session Manager is responsible for managing resources for packet sessions. It takes and processes requests from mobile station for setting up and releasing packet sessions with the PDSN.

RP: The Radio Protocol entity encapsulates Point-to-Point Protocol (PPP) frames into RP frames and tunnels them through to the PDSN.

2.2 MAC Data Plane Entity

DCR: The main functionality of the Dedicated Common Router is to send the data packet to the dedicated channel using Radio Link Protocol (RLP). The DCR is involved only in the forward direction.

RR: The Reverse Router entity is responsible for routing data packets from mobile station to the PDSN.

2.3 Priority Scheduling

Data traffic has absolute higher priority than signaling message and especially reverse direction traffic has strict higher priority than forward direction traffic. Within the control plane entities, higher priority entity starts with the order of Session Manager, Packet Session, and RP. The PCU controls the usage of CPU by serving higher priority buffer first. A lower priority buffer is preempted when a higher priority buffer is requested to serve.

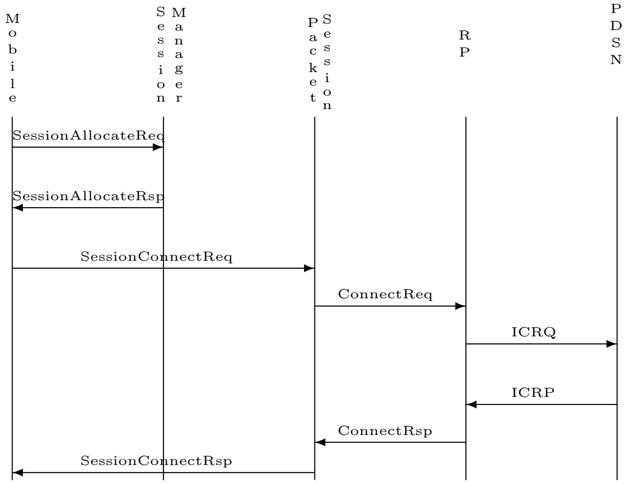


Fig. 3. Message flow of null to active or dormant to active state transition

3 MAC Layer Call Flows

This section provides MAC layer call flows with the emphasis on the PCU.

3.1 Null to Active or Dormant to Active State Transition

Fig. 3 illustrates message flows of session initiation or dormant to active state transition.

The mobile station sends *SessionAllocateReq* to the Session Manager on the PCU. This message contains CallId among other parameters. Then the Session Manager allocates a Packet Session object and responds to the mobile station providing the Packet Session Id, the signaling ACN address of the Packet Session in the *SessionAllocateRsp* message. The mobile station now sends a *SessionConnectReq* to the Packet Session. The Packet Session initiates the setup of the RP connection by sending *ConnectReq* to the RP control entity which sends an *Incoming Call Request* message to the PDSN. Then the PDSN responds with an *Incoming Call Reply* message to RP and RP indicates the successful connection setup to the Packet Session. After mobile station receives *SessionConnectRsp* from the Packet Session, PPP session is established between the mobile and the PDSN.

3.2 Active to Null or Dormant to Null State Transition

Fig. 4 illustrates message flows of active to null or dormant to null state transition.

For PPP termination with session release, the mobile station sends *SessionDisconnectReq* to the Packet Session. The Packet Session terminates the RP

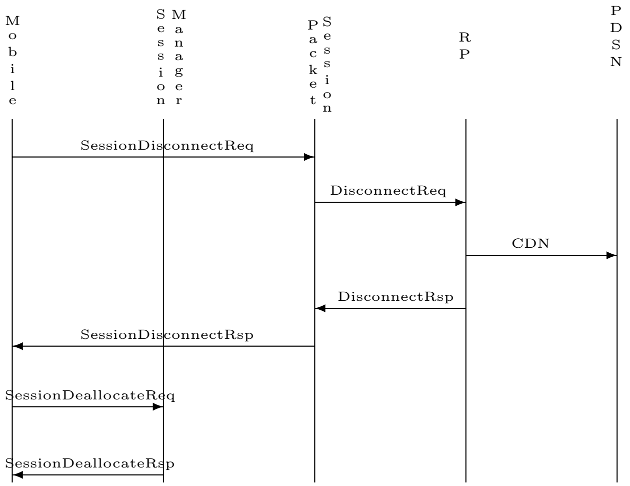


Fig. 4. Message flow of active to null or dormant to null state transition

connection to the PDSN by sending *DisconnectReq* to the RP entity which sends *Call Disconnect Notify* message to the PDSN. As soon as the *CDN* is reliably delivered to the PDSN, the RP control entity clears all references to this Packet Session's data layer entities and responds to it with *DisconnectRsp* message. The Packet Session responds to the mobile station with it SessionDisconnectRsp and transition to null state. The mobile station initiates a deallocation of the packet session object by sending *SessionDeallocateReq* message to the Session Manager which deallocates the packet session and sends *SessionDeallocateRsp* to the mobile station.

4 Simulation Setup

In our simulation, every session, for example, WWW or ftp, is assumed to correspond to a distinct mobile user. The session arrival is assumed to be Poisson distributed random variable and the session inter arrival time is exponentially distributed random variable. The average session holding time has exponential distribution with mean 180 [sec]. The active and dormant call holding time is exponential distribution with mean 20 [sec], respectively, so that the average number of active calls per session in five. During the active time, packet is arrived at every 20 [msec] for reverse traffic. The packet inter arrival time for forward traffic depends on the data transmission rate. It is assumed that the maximum data rate for forward direction is 153.6 kbps while the data rate for reverse direction is fixed to 9.6 kbps. The steady state PCU CPU utilization for high data rate traffic is assumed as given in Table 1.

The average setup delay to transit to active state is shown in Fig. 5. In this case, it is assumed that all session users use the same data rate. Since 2X calls

Table 1. Steady state PCU CPU utilization for each data rate traffic

Data Rate kbps	CPU Usage for Forward Traffic [%]	CPU Usage for Reverse Traffic [%]	Total CPU Usage [%]
19.2 (2X)	0.073	0.073	0.146
38.4 (4X)	0.164	0.073	0.237
76.8 (8X)	0.3	0.073	0.373
153.6(16X)	0.66	0.073	0.733

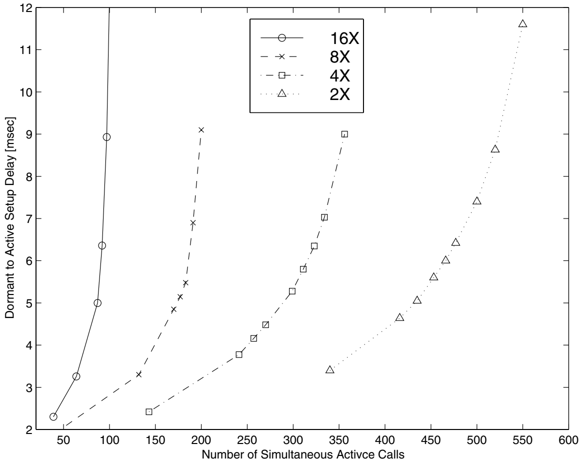


Fig. 5. Average null to active or dormant to active setup delay with respect to simultaneous active calls: data rate is fixed to certain rate

utilize less CPU than higher data rate calls, it is allowed more simultaneous 2X calls before the call setup delay increases abruptly. With the CPU defined in Table 1, the number of allowable simultaneous active calls is 80, 160, 310, and 470 for 16X, 8X, 4X, and 2X, respectively. If the number of active call in the PCU is above those numbers, the average setup delay is significantly increased and it reached beyond the upper limit of allowable setup delay in QoS point of view.

In Fig. 6, the data rate is mixed based on a certain distribution which is assumed that 6% for 16X, 8% for 8X, 10% for 4X, and 76% for 2X calls. The average setup delay is abruptly increased at the utilization factor, the ratio of active call arrival rate in the PCU to PCU service rate, is about 0.85 and the simultaneous number of active call is about 350.

5 Conclusion

In this paper, the average null to active or dormant to active setup delay was observed for PCU which is a new component to serve high speed packet data

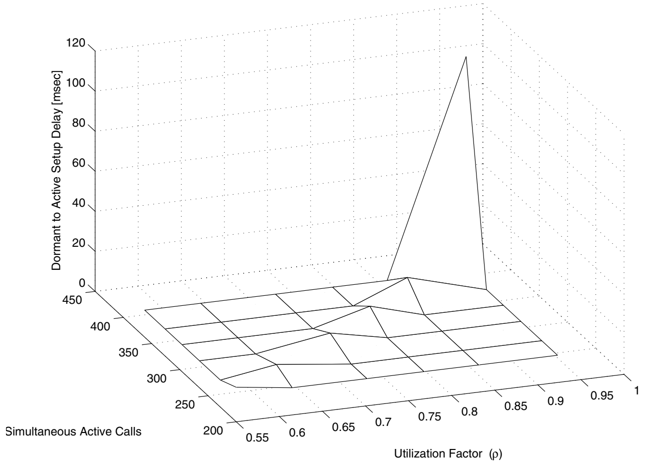


Fig. 6. Average null to active or dormant to active setup delay with respect to simultaneous active calls and utilization factor: data rate is mixed

traffic. As the demand of high data packet usage is increased, it is recommended to use faster CPU to process large number of simultaneous active calls. The flow control and congestion control study for MUX device shown in Fig. 1 are important to satisfy different QoS requirements for voice and data, less voice packet delay and less data packet loss.

References

1. TIA/EIA/IS-2000-3: Medium Access Control (MAC) Standard for cdma2000 Spread Spectrum Systems

Increasing the Testability of Object-Oriented Frameworks with Built-in Tests

Taewoong Jeon[1], Sungyoung Lee[2], and Hyonwoo Seung[3]

[1] Dept. Computer & Information Science, Korea University, Korea
jeon@selab.korea.ac.kr

[2] Dept. Computer Engineering, Kyunghee University, Korea
sylee@oslab.kyunghee.ac.kr

[3] Dept. Computer Science, Seoul Women's University, Korea
hwseung@swu.ac.kr

Abstract. Object-oriented frameworks require thorough testing as they are reused repeatedly in developing numerous applications. Moreover, frameworks must be retested each time they are adapted and extended for reuse. Frameworks, however, have properties that make it difficult to control and observe the testing of the parts that were modified or extended. This paper describes a scheme of encapsulating test support code as built-in test (BIT) components and embedding them into the hook classes of an object-oriented framework so that defects caused by the modification and extension of the framework can be detected effectively and efficiently through testing. The test components built into a framework in this way increase the testability of the framework by making it easy to control and observe the process of framework testing without incurring changes or intervention to the framework code.

1 Introduction

One of the current trends in object-oriented technology is to develop software as a framework[1,2,3]. A framework supports efficient software development by providing a pre-implemented architecture common to a family of applications together with hot spots designed to be adapted to each particular application. An object-oriented framework consists of various cooperating abstract and concrete classes[4]. Some of the classes in the framework are designed as hook classes to serve as the hot spots that can be adapted and extended within the limits the architecture permits. That is, the framework is reused in application developments by adapting and extending the hot spots provided as hook classes according to the class inheritance and object composition mechanism.

Since frameworks are to be reused repeatedly in many application developments, they need thorough testing. Moreover, each time a framework is extended for reuse, it requires additional testing to check for possible progressive faults and regression faults. Systematic testing on a framework is therefore crucial for the reliability of framework-based applications. Frameworks, however, have properties that make it difficult to control and observe the process of framework

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 169–182, 2002.

testing needed whenever they are modified and extended for reuse in developing applications.

Usually, the execution of the extended parts of the framework is controlled by the framework itself, which makes it difficult to set up initial test conditions of the framework and to drive test execution. Since it is not easy to predict the starting point of the execution and to observe the result of the test, it is also difficult to detect occurrences of malfunctions. Consequently, it is not easy to force the classes adapted and composed for reuse to satisfy the constraints the framework assumes, or to discover constraint violations in advance. If changes are made to framework code arbitrarily to intermingle it with test support code to enhance the controllability and observability of framework testing, the reliability of testing can be compromised due to possible interferences of test code with the framework code.

In order to overcome such problems, this paper proposes a scheme for embedding test support code as BIT(Built-in Test) components into the hook classes of the framework without incurring changes or intervention to the framework code. The test components built into the framework in this way make the testing of the adapted framework more controllable and observable, and thereby enable us to effectively detect, through tests, the faults generated during framework adaptation or extension.

This paper is organized as follows. Sect. 2 reviews related work. Sect. 3 and Sect. 4 discuss the testability and hot spots of a framework, respectively. Sect. 5 presents our proposed design scheme for embedding built-in test components in framework hot spots. Sect. 6 describes a case study in which we applied the design scheme to the testing of a sample framework. Sect. 7 concludes this paper.

2 Related Work

Many testing methods for object-oriented software have been proposed[5,6,7,8,9,10,11,12]. For example, ASTOOT[7] offers an algebra-based class test method and support tools, while ClassBench[8] provides a state-based class test method and its support environment. An incremental test for the class hierarchy[9] and object-oriented integration testing methods[10] have been also introduced. Binder[6,11] comprehensively presented test design patterns and methods to construct a test support environment for object-oriented software. The XUnit[12] provides testing frameworks for many programming languages to support writing repeatable unit tests of classes. For example, the JUnit[13] framework, which is a version of XUnit framework for Java, can be extended to implement a suite of test cases for a Java class, and any other test support code necessary to automate running the test cases. Edwards[14] proposed a strategy for automated generation of test drivers, test cases, and test oracles as BITs for components (or classes) given their specifications. As with our testing approach, both the XUnit's and Edwards' strategies separate the testing infrastructure code from the units under testing. Difficulties in testing adapted frameworks are well-known, and several solutions have been proposed in the literature[6,15,16,17]. Fayad, et al.[17] presented a method in which the test-case generating codes, packaged in Built-in tests(BITs) classes, are embedded into the framework, and reused in framework testing, being inherited and adapted during framework adaptation and extension.

Although all of the testing approaches and methods mentioned above are useful for framework testing, they do not address explicitly the testing problems specific to the hot spots of a framework in which modifications are made whenever the framework is reused for developing applications. Our approach differs from, and complements the previous work in that it focuses directly on increasing the testability of the variable parts of a framework that need more frequent, and so, more efficient testing than other parts of the framework.

3 Framework Testability

Software testability means ease of revealing software faults through tests[11,18,19]. For effective framework testing, high testability must be maintained when developing, adapting or extending a framework. Software testability can be affected directly or indirectly by many factors[11]. In this paper, we focus on the following four factors that have direct influence on framework testability:

1. controllability: the ability to set up and control test conditions
2. sensitivity: the ability to capture and expose traces of malfunctions in response to tests
3. observability: the ability to observe test results externally
4. oracle availability: the ability to determine or obtain expected test results

Besides testability, there are other important quality factors such as reliability, robustness, flexibility, modularity, and performance that a framework must maintain at a high level. Though testability is generally complementary to other quality factors, it could conflict with them in some cases. For example, increasing a framework's test sensitivity could cause malfunction by faults to occur more frequently, and consequently to reduce reliability and robustness. To take another example, if a framework's controllability and observability get higher, then reliability, flexibility, modularity or performance could be degraded since concealed information could be revealed and components of the framework might be possibly interfering with each other.

In order to increase framework testability without adverse effects on other quality factors, we designed a scheme for encapsulating test support codes as a set of tester components separated from the framework under testing and embedding them into the framework with little change to the framework. Since test support codes are embedded as BIT components with little interference to the framework under test, enforcing testability does not sacrifice other quality factors such as reliability and modularity. Furthermore, the tester components are designed to be attached or detached as needed so that framework testability can be higher during tests and lower in operation in order to avoid performance degradation due to the overhead incurred by the execution of tester components.

4 Framework Hot Spot

A framework is composed of frozen parts designed to be shared among applications without modification, and hot spots designed to be adapted to specific needs of the application[20,21]. A framework also prescribes rules of composition and interaction among the system components which must be observed when adapting hot spots. Those rules can be defined precisely using the design by contract principle[22,23,24].

The frozen and hot spots of an object-oriented framework are encapsulated in methods of classes. The method for a frozen spot is called a template method, and the method for a hot spot is called a hook method[20,21]. The class that contains template methods is called a template class, and the class that contains hook methods is called a hook class. The template class and the hook class can be of different classes or the same class. When they are different, the hook class is adapted and extended by composing an object from a subclass of the hook class into the template class through an instance variable which references the hook class object. When they are same, the hook method is adapted and extended by subclassing the unified template/hook class.

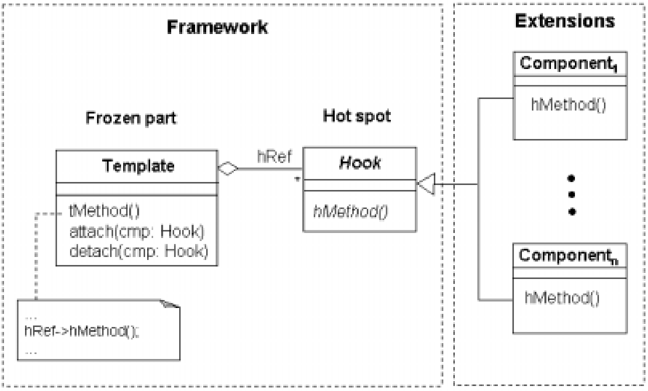


Fig. 1. Framework Hot Spot

Pree classified the composition among the template and hook classes of a framework into 7 patterns[20]. An application built from an object-oriented framework provides its functions through interactions allowable among the template and hook class objects compounded according to the composition patterns. The variable functions implemented through adapting and extending the hook classes, and the interactions among the frozen spots and extended hot spots of the framework must be (re)tested to check for possible progressive faults and regression faults. Fig. 1 shows a framework hot spot in which a template class is associated with its hook class that can be extended into subclasses as represented by Component$_i$.

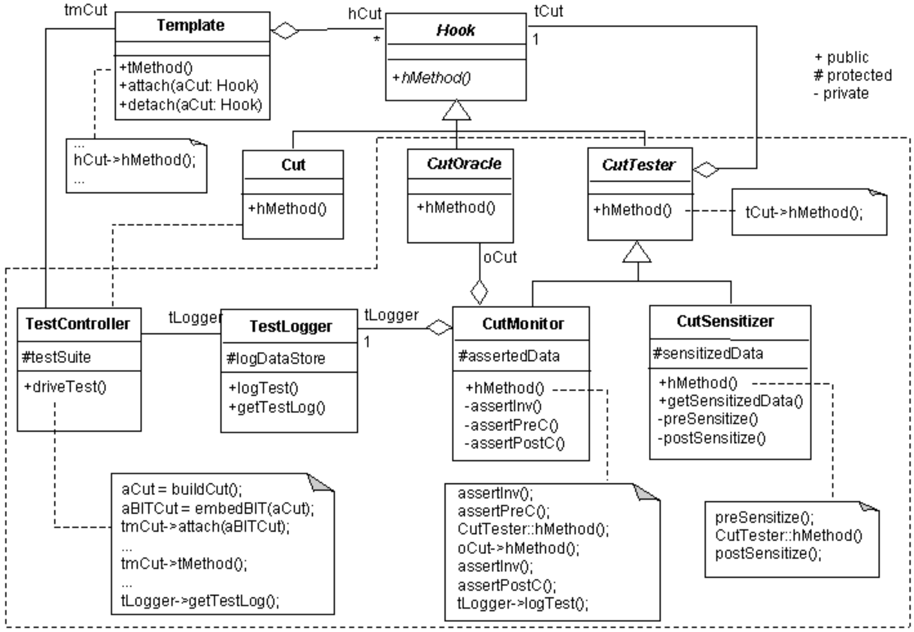


Fig. 2. Class Structure of BIT-Embedded Framework Hot Spot

5 The Design Scheme of BIT-Embedded Framework Hot Spot

This section describes the proposed design scheme for embedding tester components into the hook classes of the framework in order to facilitate testing whether the framework's functional contracts are observed when the hook classes are extended to adapt to an application. The design scheme is illustrated using the UML diagramming notations and the C++ language.

The types of test support components we devised to enforce framework testability are test controllers, test sensitizers, test monitors, test oracles, and test loggers. The test controller increases controllability by setting up and initializing test conditions for framework hot spots. The test sensitizer increases the framework's sensitivity to faults by capturing and leaving traces of malfunctions during test execution. The test monitor increases observability by monitoring actual results of test execution and judges pass/fail of the test by comparing the actual with the expected results. The test oracle helps the monitor judge pass/fail of the test by giving expected pre-conditions, post-conditions and invariants of the test cases. The test result, passed or failed, is sent to the test logger by the monitor. The test logger records and stores the test result according to the current test context.

The class diagram in Fig. 2 shows the composition pattern of a BIT-embedded framework hot spot in which the tester components are embedded into a hook class of the framework. Subclasses that extend the hook class is the framework CUTs (classes or components under test). The classes in the dotted area of the diagram represent the tester components embedded or attached to the framework hot spot. The TestController and TestLogger are the tester components attached from the outside of the CUT, while the other components are BITs embedded into the CUT. The CutTester is an abstract base class of the BIT components which provides the same interface as the CUT for the template class object through the hook class. The CutSensitizer and CutMonitor are designed as subclasses of the CutTester class, which is a subclass of the hook class. The BITs and the CUT form a chain structure connected through the tCut, which is an instance variable of the CutTester class. The Template object is composed with the CUT through the Hook class interface referenced by the hCut variable. Via a chain of the BIT components in between, the template object is connected to the CUT located at the end of the chain.

When its hMethod is called, each of the tester components embedded into the CUT as elements of the chain structure performs its own testing function before and after forwarding the hMethod call to the successor component connected through the tCut variable until it reaches the CUT. During the process, however, the functional behavior between the Template object and the CUT is not interfered with by the tester components attached to the CUT. The design pattern of the tester components is similar to the Decorator design pattern[4]. Testing functions on the CUT are distributed over, and encapsulated by the tester components. The outermost CutMonitor plays the role of a proxy which watches the template object have access to the CUT[4].

The CutSensitizer in Fig. 2 captures the clue data critical to the testing of the CUT before and after it calls the hMethod through tCut. For example, the CutSensitizer keeps the state before the CUT is called, so that it is possible to inspect post-conditions and invariants after calling. The CutMonitor checks if the contract is violated by inspecting the pre-, post-conditions, and invariants before and after the CUT calls the hook method. The test results are collected and recorded by the TestLogger.

The CutOracle referenced by the CutMonitor calculates the expected result after the CUT executes the hook method. The CutMonitor judges pass or fail of the test by comparing the expected result obtained from the CutOracle with the actual result obtained from the CutSensitizer.

The TestController initializes the other tester components and drives the test execution. Before starting the test execution, it generates an object structure instance of the CUT and BIT components, and embeds the generated BITs in the CUT instance. It then initializes the BIT-embedded CUT and the TestLogger to the condition required by the current test case.

Fig. 3 shows an object structure which has a test monitor, sensitizer and oracle between the template object and the CUT object instantiated according to the BIT-embedded composition pattern in Fig. 2. Fig. 4 shows a sequence of interactions among the template, CUT and tester objects of Fig. 3 when the tMethod of the template object is called by the TestController.

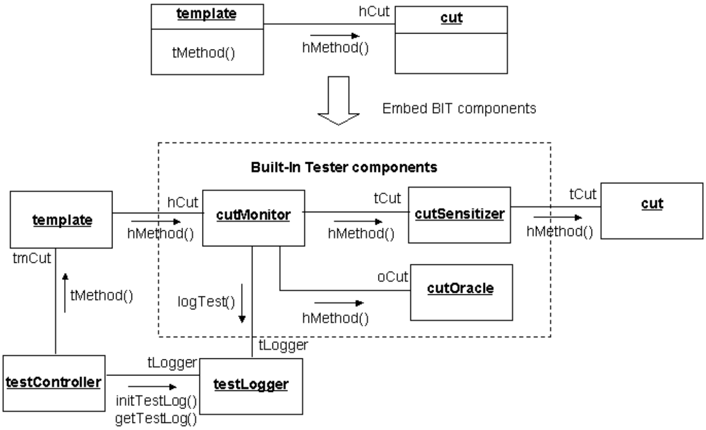


Fig. 3. Object Structure of BIT-Embedded Framework Hot Spot

Referencing the hCut variable, the tMethod of the template object calls the hMethod of the hook class object. The invocation of the hMethod is forwarded to hMethod of the CUT by way of the CutMonitor and the CutSensitizer. The execution result returns to the tMethod of the template object by way of the CutSensitizer and the CutMonitor. With the help of the CutOracle, the CutMonitor checks the pre-, post-conditions and invariants of the CUT before and after the hMethod of the CutSensitizer is called, and passes the inspection result to the TestLogger. The CutSensitizer captures and stores the states of the CUT before and after the hMethod of the CUT is called. During this process, however, the interaction between the template object and the CUT is not interfered with by the CutMonitor, CutSensitizer and CutOracle.

The object structure of the hot spot in Fig. 3 and Fig. 4 shows a 1:1 composition of a hook class object and a template class object. The BIT embedding method proposed in this paper supports the framework testing for 1:n composition where a template object is compounded with many hook class objects, as well as for 1:1 composition. For example, in the object structure allowed by Fig. 2, where a template object is 1:n compounded with many hook objects, the BIT components are embedded into each of the hook objects. Although this paper exemplifies a case with only one hook method, testing can be easily extended to the interactions between a template object and a hook object that has several hook methods. In such a case, the BIT-embedding process is defined for each hook method in the same way as the BIT components are embedded in the hMethod in Fig. 2.

6 A Case Study

This section describes a case study we performed to justify our approach. The framework exemplified here is an alarm monitoring framework that we imple-

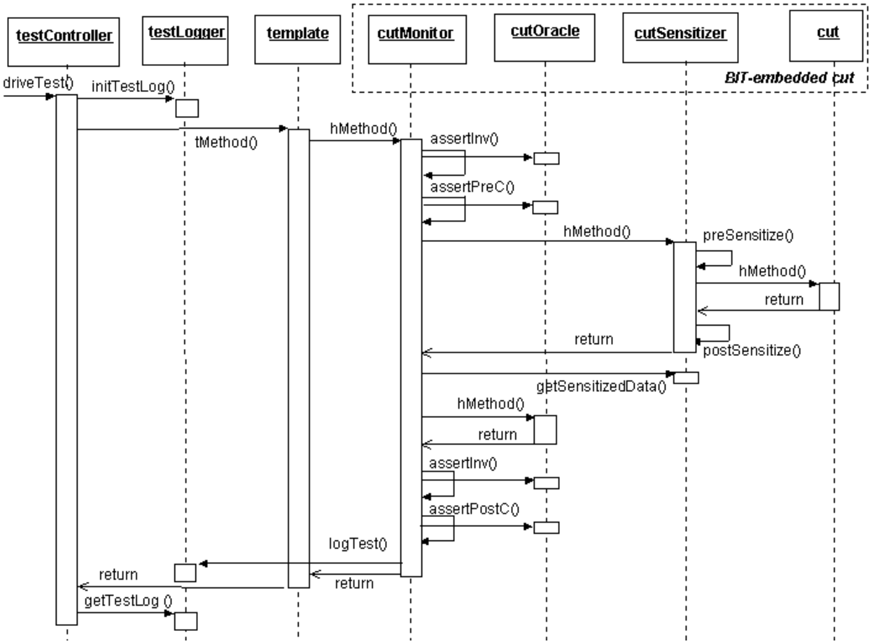


Fig. 4. Sequence Diagram of BIT-Embedded Framework Hot Spot

mented in C++ for use in applications such as process control monitoring systems. The framework can be extended to an alarm monitoring system which raises an alarm upon change to an abnormal state according to the current measurement and configuration.

The class diagram in Fig. 5 shows a part of the composite structure of the alarm monitoring framework. The InPoint is the class which directly inputs measured values from the outside. The DerivedPoint class calculates new measured values or state values, using the measured values acquired from other Point class objects. AlarmMonitor, a subclass of the DerivedPoint class, senses state changes, depending on the changes of measured values, and gives an alarm upon change to an abnormal state. AlarmMonitor is a hook class compounded into the Point class, and can be extended to various subclasses depending on the applications.

The AlarmMonitor will be the CUT to which the tester components are attached in the example framework, and have the state behavior which is specified in the Statechart[25] shown in Fig. 6. If the measured values received as parameters when the update operation is called exceed either high or low limits, the AlarmMonitor raises high or low alarm. It also has deadbands to avoid repeating alarms that might occur when the measured values fluctuate between upper or lower boundaries. Fig. 7 shows one possible sequence of interactions between the objects of an alarm monitoring system when a sequence of measured values comes from a single input source.

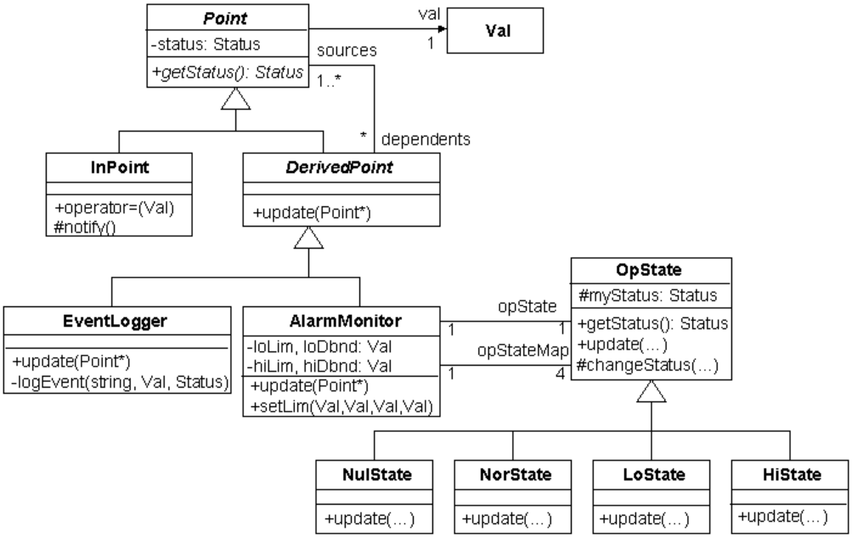


Fig. 5. Partial Class Structure of the Alarm Monitoring Framework

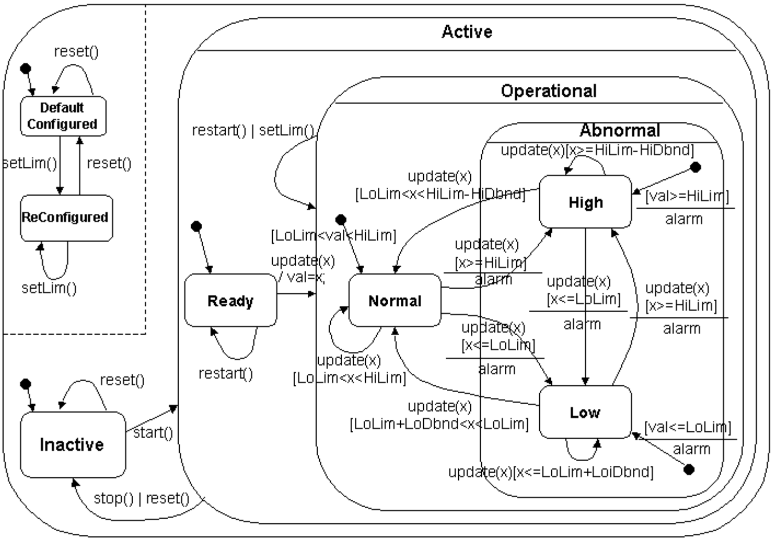


Fig. 6. State Behavior of the AlarmMonitor Class

Fig. 8 shows a class structure in which tester components, as BITs, are embedded into the CUT, AlarmMonitor. The classes in dotted area of the diagram indicate the embedded BITs, which are CutMonitor, CutSensitizer and CutOracle. The CutMonitor and the CutSensitizer are connected to each other through tCut, an instance variable of their superclass, CutTester. The CutSensitizer in-

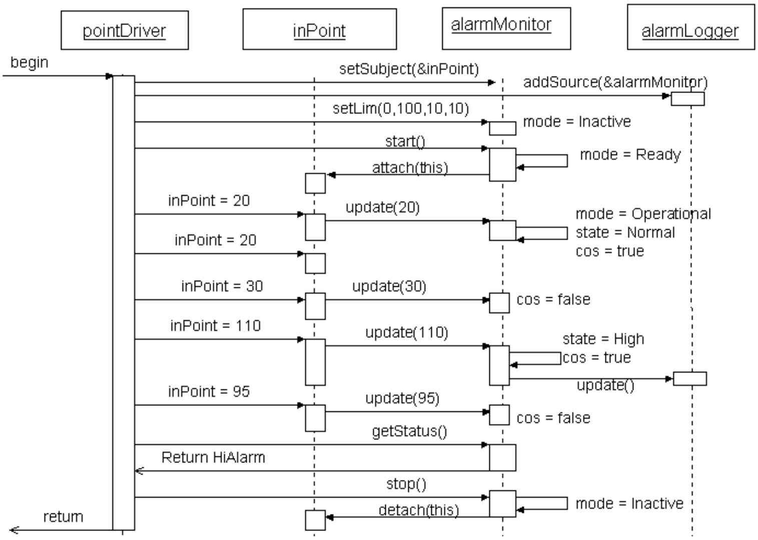


Fig. 7. Sequence Diagram of the Alarm Monitoring System

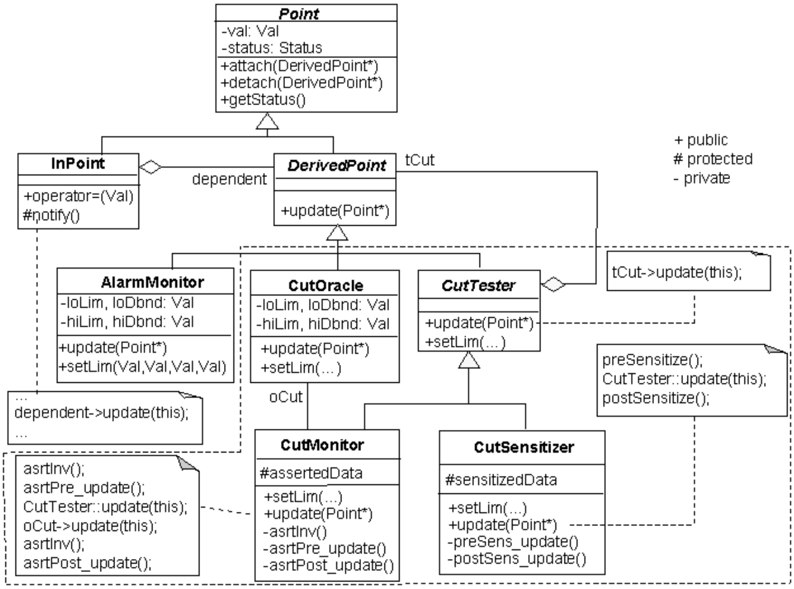


Fig. 8. Class Structure of BIT-Embedded Alarm Monitor

spects and captures the internal states of the AlarmMonitor before and after it calls the update function through tCut. The CutOracle is designed to simulate the state behavior in Fig. 6. The Cut Monitor is designed to check if the CUT

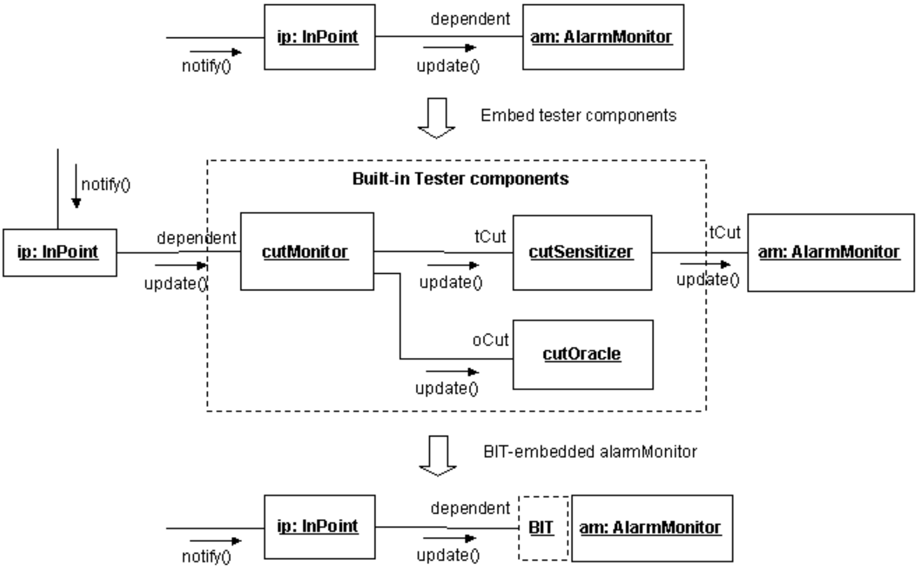


Fig. 9. Embedding BIT Components in an AlarmMonitor

violates the pre-, post-conditions and invariants, comparing the actual behavior from the CutSensitizer with the expected behavior from the CutOracle. Other tester components, such as a test controller which initializes test conditions and drives the framework testing, or a test logger which collects and records the test results, are also designed to be attached to the framework under testing. As test cases, we used sequences of update function calls whose parameters are the measured values. Each of such test cases represents one of the paths of a state transition spanning tree whose paths are the state transition sequences allowable within the operational mode in the statechart in Fig. 6.

At the initial stage of the testing, the test controller instantiates the tester components and embeds them into the CUT. When the test controller attaches an AlarmMonitor to the InPoint object, the BITs embedded in the AlarmMonitor are also attached together with the AlarmMonitor. The tester components were implemented in such a way that they were integrated with the framework under testing without incurring changes or intervention to the framework codes. The tester components can be dynamically detached and attached as needed at run-time.

Fig. 9 shows how the tester component objects, designed according to the pattern in Fig. 8, are embedded into the AlarmMonitor object which is compounded as a hook class object into the InPoint object. Upon testing, not only establishing test conditions but also monitoring test results are done transparently without intervening in the alarm monitoring function of the system.

In order to check if the tester components work properly in testing the BIT-embedded alarm monitoring system, we planted some errors in the system that

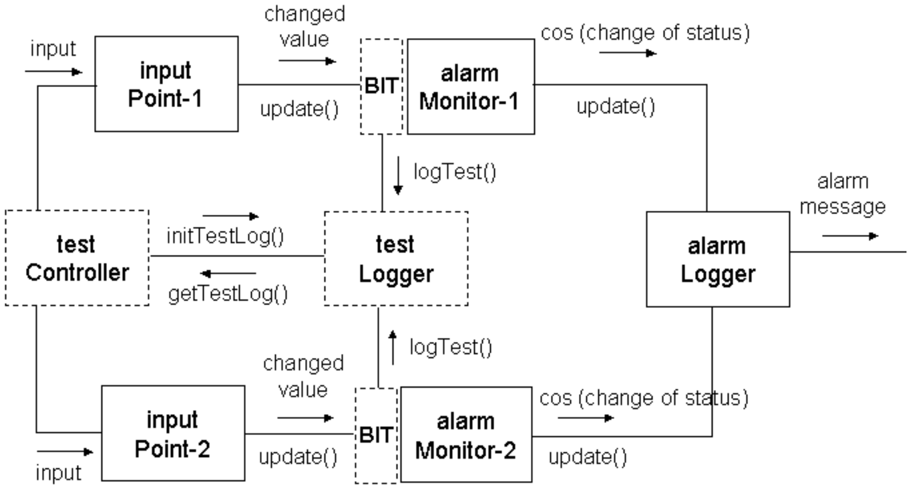


Fig. 10. Possible Object Structure of BIT-Embedded Alarm Monitoring System

may not be easily detected but cause abnormal state changes, and performed testing with test patterns as explained in the previous section. Two cases of testing have been done: with and without embedding BITs. In the case of tests with BITs, abnormal state changes caused by errors were recorded by the test logger without any omissions, though not all the known errors were detected. Meanwhile, in the case of tests without BITs, only a subset of the errors that could be detected by the BITs, such as giving a false alarm by misjudging a normal state to be abnormal, were detected by the alarm logger.

Though Fig. 9 illustrates only a simple case in which a single input and a single alarm monitor are involved, we did successfully apply our testing scheme to more complicated cases in which multiple input sources and multiple alarm monitors were involved in a variety of configurations. Fig. 10 shows one of such cases in which each alarm monitor has its own set of built-in test components except for the test controller and the test logger that are shared by the system.

7 Conclusion

This paper has described a scheme for increasing the testability of an object-oriented framework by encapsulating test support code as built-in test (BIT) components and embedding them into the hook classes of the framework. Test components built into a framework in this way make it easy to detect defects caused by the modification and extension of the framework through testing. Using our scheme, the test components can be attached and detached to/from the framework at run-time as necessary without incurring changes to the framework code and without affecting its functional behavior. We showed how the proposed testing scheme worked for testing of a particular framework. The example framework with which we performed the case study was, however, a small one compared with the real-world frameworks in actual use. We plan to elaborate our testing scheme with more case studies so that it can scale up well for testing larger frameworks.

Acknowledgements

This work was partially supported by the Basic Research Program of the Korea Science & Engineering Foundation under Grant No. R01-1999-00238.

References

1. Fayad, M.E., et al.: Building Application Frameworks, John Wiley & Sons (1999)
2. Fayad, M.E., et al.: Implementing Application Frameworks, John Wiley & Sons (1999)
3. Fayad, M.E. and Johnson, R.E.: Domain-Specific Application Frameworks, John Wiley & Sons (2000)
4. Gamma, E., et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley (1995)
5. Kung, D.C., et al. (eds.): Testing Object-Oriented Software, IEEE CS Press (1998)
6. Binder, R.V.: Testing Object-Oriented Systems: Models, Patterns, and Tools, Addison-Wesley (2000)
7. Doong, R. and Frankl, P.: The ASTOOT Approach to Testing Object-Oriented Programs, ACM Trans. Software Eng. and Methodology, **3** (1994) 101–130
8. Hoffman, D. and Strooper, P.: ClassBench: a Framework for Automated Class Testing, Software Maintenance: Practice and Experience, **27** (1997) 573–597
9. Harrold, M.J., et al.: Incremental Testing of Object-Oriented Class Structures, Proc. 14th Int'l Conf. Software Eng. (1992) 68-80
10. Jorgensen, P.C. and Erickson, C.: Object-Oriented Integration Testing, Comm. ACM, **37** (1994) 30–38
11. Binder, R.V.: Design for Testability in Object-Oriented Systems, Comm. ACM, **37** (1994) 87–101
12. The XUnit Home Page, http://www.xprogramming.com/software.htm
13. Gamma, E. and Beck, K.: JUnit A Cook's Tour, Java Report (1995)
14. Edwards, S. H.: A Framework for Practical Automated Black-Box Testing of Component-Based Software, Software Testing, Verification and Reliability, **11** (2001) 97–111
15. Codenie, W, et al.: From Custom Applications to Domain-Specific Frameworks, Comm. ACM, **40** (1997) 71–77
16. Sparks, S, et al.: Managing Object-Oriented Framework Reuse, IEEE Computer, **29** (1996) 52–61
17. Fayad, M.E., et al.: Built-In Test Reuse, In the Building Application Frameworks, Fayad, M.E., et al, John Wiley & Sons (1999) 488–491
18. Voas, J.M., et al.: Predicting Where Faults Can Hide from Testing, IEEE Software (1991) 41–48
19. Voas, J.M. and Miller, K.W.: Software Testability: The New Verification, IEEE Software, **12** (1995) 17–28

20. Pree, W.: Design Patterns for Object-Oriented Software Development, Addison-Wesley (1995)
21. Schmid, H.A.: Systematic Framework Design by Generalization, Comm. ACM, **40** (1997) 48–51
22. Meyer, B.: Applying Design by Contract, IEEE Computer (1992) 40–51
23. Helm, R, et al.: Contracts: Specifying Behavioral Compositions in Object-Oriented Systems, Proc. OOPSLA'90 (1990)
24. Steyaert, P, et al.: Reuse Contracts: Managing the Evolution of Reusable Assets, Proc. OOPSLA'96 (1996)
25. D. Harel, et al.: STATEMATE: a Working Environment for the Development of Complex Reactive Systems, IEEE Trans. Software Eng., **16** (1990) 403–414

Guaranteeing the Continuous Stream Service in Cluster Media Servers

Sungin Jung[1], Hagyoung Kim[1], and Cheolhoon Lee[2]

[1] Computer System Division
Electronics and Telecommunication Research Institute,
Daejon, Korea
{sijung,h0kim}@etri.re.kr
[2] Parallel Processing Laboratory,
Department of Computer Engineering,
Chungnam National University, Daejon, Korea
chlee@ce.cnu.ac.kr

Abstract. Due to the explosive growth of the Internet, stream services are becoming more popular. The large media servers supporting thousands of concurrent media streams have to satisfy us with both good throughout and high availability. Although they are reciprocal from the viewpoint of system performance, the stream service continuity, as well as throughput, should be considered in clustered media servers. This is because media stream services should be continuously available to the clients even on the event of system failure. In this paper, we propose *LSS* and *RRD* mechanisms for the purpose of guaranteeing continuous stream services. These mechanisms provide additional functionalities for the traditional media servers. This paper focuses on how to apply these mechanisms to clustered servers with the minimum overhead according to the cluster size.

1 Introduction

A wide range of applications in entertainment, business, and education require the development of scalable multimedia servers that can support efficient means of storing and delivering digital audio and video objects to many clients. For example, video-on-demand applications require supporting thousands of concurrent video streams and storing a large number of video objects. Clustered multimedia servers consisting of a set of processing nodes, each with a local disk array, connected by a high bandwidth network, have been proposed [1,2,3] to provide a scalable solution that meets the requirements of concurrent media streams.

In general, the basic requirement for any continuous media servers is the delivery of isochronous streams. A second key requirement is providing high availability and continuous delivery in the presence of failures. Availability is more critical in a scalable server such as a clustered system. However, as the number of components (nodes and disks) in the clustered systems increases, the

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 183–193, 2002.

probability of system failure increases, resulting in the outage of media stream services to clients. In conventional clustered systems, when a system failure occurs, the clients have to wait until the services resume. This is unacceptable for continuous media applications like as video-on-demand and Internet broadcasting. These applications require that media contents should be continuously available to the clients even on the event of system failures.

Consider the example of a home movie theater. In the event of a system failure, it would be certainly more desirable to watch the movie continuously within a short time than to stare at the blank screen. To do this, it is necessary that streams from the failed nodes must be reassigned to some healthy nodes with minimum service disruption [4]. Component faults causing the overall system failure can roughly be classified into the storage system faults and the service delivery system faults. Most of current works in this field address the high availability of storage systems because it causes a system-wide impact and plays an important role in media contents management. However, this alone cannot guarantee the overall service continuity. To guarantee the service continuity, we should address not only the availability of storage systems to deal with storage system faults, but also continuous stream services to deal with service delivery system faults. In a popular mobile Internet, the mobility and the limitation of power and resources in mobile devices could result in frequent service disconnection during stream services.

This paper proposes three mechanisms for guaranteeing continuous stream services in the event of system failures happened during service delivery, and investigates the impact on the performance caused by using them. The basic concept of our mechanisms is the management of status, which is used to resume the closed service. It is important how to exactly resume the closed service within a short time. These mechanisms provides additional functionalities for the traditional media servers, so that our main focus of this paper is to find out how to apply these mechanisms to clustered servers with a minimum overhead according to the cluster size.

The rest of the paper is organized as follows. In Section 2, we describe the previous works related to this research project. Section 3 describes a general clustered architecture for media servers. Section 4 introduces the proposed mechanisms for guaranteeing service continuity, and formally describes the goal of our mechanism. Simulation results are presented and analyzed in Section 5. Finally, we present the conclusions and discussions for the future work.

2 Related Works

Most of the previous works on high availability in media servers focus on storage systems because the media contents stored on them are the most important resources in the media streaming service. Some of them [7,8,9,10] focused on RAID schemes across disks within a single server, while others [3,11] addressed disk layouts for real-time multimedia applications without considering high availability. So as to guarantee quality of service, we have to address real-time issues as well

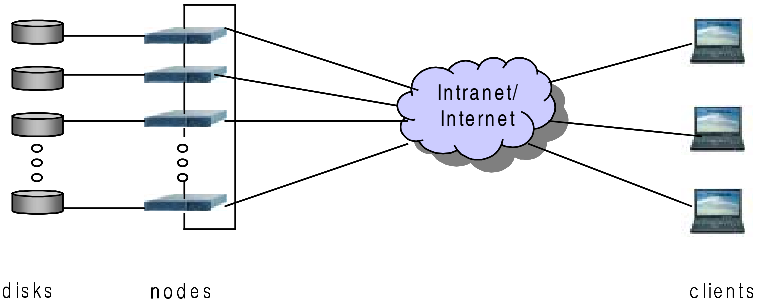


Fig. 1. Clustered Server Architecture

as high availability in stream services. Tewari et. al. [5] studied mirroring and software RAID schemes with different placement strategies that guarantee high availability in the event of disk and storage node failures on clustered system, while satisfying the real-time requirements of media streams.

Recently there have been some researches for designing a failure and overload tolerance mechanism for continuous media servers [4]. They proposed solutions for graceful recovery from overload scenarios arising out of server failure or customer interactions, but they did not consider the recovery mechanism to resume the closed service. In other words, they focused only on the overload management of media streaming service in the event of failure. Another work [12] related to this paper investigated the suitability of a clustered architecture for designing scalable multimedia servers. Specifically, it evaluated the effects of both architectural design of cluster and read-ahead buffering and scheduling on the real-time performance provided by the server.

3 Clustered Server Architecture

A clustered server architecture, illustrated in Fig. 1, consists of a set of nodes connected by a inter-cluster network. Here we briefly outline the architecture aspects required in the subsequent sections of this paper. Each node has a set of local disks attached to it. The nodes of a clustered architecture can be divided into two logical categories: delivery nodes and storage nodes. The clustered server can be configured to behave as a flat architecture or a two-tier architecture [3]. In the flat architecture, each node provides functionalities of both storage node and delivery node. However, in the two-tier architecture, the nodes are partitioned into two categories, respectively. In either architecture, the delivery nodes send media streams stored in their disks to clients through Intranet or Internet.

In this paper, we assume the high availability techniques for disks such as software RAID [5]. The failure of a delivery node will result in the loss of all streams served by that node. All streams using failed node for delivery will lose their connection, so that these streams should be resumed from another delivery node after a new connection is established. In this case, the service continuity is not guaranteed

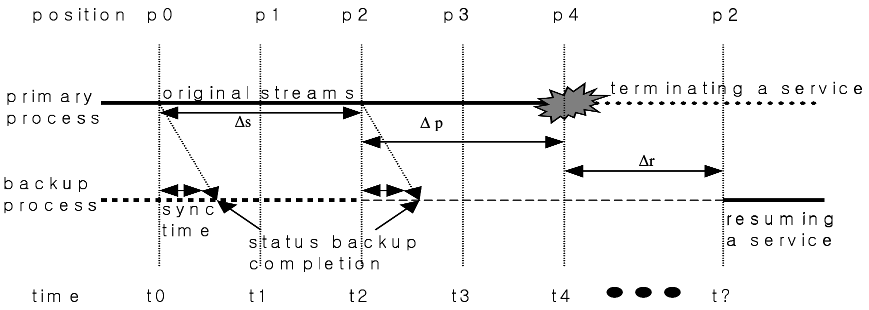


Fig. 2. Continuous Service on Failure

4 Proposed Mechanisms for Service Continuity

This section describes the proposed mechanism for continuing media stream services on the failure of delivery node at a clustered architecture. The purpose of our proposal is that closed services resume within a short time from a near stream position terminated on the failure.

4.1 Schemes for Service Continuity

In order to guarantee continuous playout, the failure of delivery node and storage node including disks must be considered. In case of the failure of storage system, its availability scheme was studied in the previous papers [3,6,7,8,9,10]. The typical MTTF (Mean Time To Failure) value for a single node is around 18,000 hours. This means that the MTTF of some node in a ten-node clustered system is on the order of 1,800 hours (approximately 75 days). Handling node failures becomes increasingly important because they not only occur more frequently but also cause more disruption to service [8,13]. The key concept of the proposed scheme guaranteeing continuous services at delivery node is status monitoring, by which we can decide the position to resume the stream services during failure recovery.

In Fig. 2, at time t3, a primary process has carried the video stream of position p2 to a client, and a backup process kept up the streams delivery status (position p2). At time t, let $P_{primary}$ and P_{backup} be the stream positions of the primary and backup processes, respectively. Then the equation, $P_{backup} < P_{primary}$, is always satisfied because the backup stream follows the primary stream. The synctime is the time taken to finish status synchronization, and it depends on the size of status information and network capacity. Assume that a fault occurs on a delivery node at time t4 before the completion of status synchronization. Note that the stream position of the backup process is p2. Also assume that the status is synchronized at every two-stream period ($\triangle s$). Then there is a position difference ($\triangle p$) between the primary and backup streams. This means that the client gets the stream between p4 - $\triangle p$ to p4 positions twice, once by the primary before the failure and once again by the backup process after recovery from the failure. To reduce $\triangle p$, we have to reduce the synchronization period $\triangle s$. By reducing the period, we can provide more smooth service continuity for clients, but additional system overload is required. Also, since it is not desirable for clients to stare at the blank screen for long time, the resumption time ($\triangle r$) should be reduced. This paper suggests the mechanisms guaranteeing continuous stream services with the objective of minimizing the resumption time. The proposed mechanisms provide additional functionalities to the traditional media servers and are explained below. The main focus of this paper is how to apply the proposed mechanisms to clustered servers with the minimum overhead according to cluster sizes.

LSS (Log Streams Status) Mechanism. LSS is a technique that logs the status of streams that were already sent via each delivery node. So as to apply the LSS mechanism, we assume that there is a fault-tolerant node called the log node in the clustered server, which logs the status of streams from each delivery node. Among the status information are the movie title, the client identification, the stream position, and so on. Among them, the stream positions should be updated to the log node at every synchronization.

For example, let us assume that there are ten processes that deliver MPEG-2 movies (90 minute) into ten clients at 4.5 megabits per second. Then, the log node gets 54,000 updates for ten movies. If the synchronization period is very short (i.e., $\triangle p$ is very low) or the client requests are overflowing, the number of log updates becomes extremely high. This phenomenon brings a serious bottleneck to the log node. Even though this mechanism is simple, it takes a long resumption time ($\triangle r$) for another process to resume the closed service, since the initial operations such as process creation, resource allocation, and media accesses should be redone.

RRD (Ready Resource for Instant Delivery) Mechanism. In this mechanism, whenever a primary process starts a stream service, a backup process is created on another node with the purpose to takeover the service when a fault occurs in the node where the primary process runs. Let us assume that the number of backup process for one primary process serving a normal stream service is one in this paper. Considering the above example again, there are twenty processes during ten movies playout. The primary process forwards the status information to its backup at every synchronization. With this scheme, the backup process is always ready to resume the stream service when the primary process fails. By doing this, compared with the LSS mechanism, the RRD mechanism can reduce the resumption time ($\triangle r$) dramatically.

We propose two methods for the backup process to prepare media contents to resume the stream service. With the first one called *RRD-disk*, the backup process uses its local disks on the assumption that all media contents are in the disks of all nodes. Therefore, the RRD-disk requires at most two times as much as normal IO operations to IO subsystem. In the second method called *RRD-*

net, the primary process forwards the media contents instead of the status to its backup. By doing this, the RRD-net method be able to resume the stream service directly from the latest contents. The RRD-net brings overload to the network subsystem. In the above example, let us assume that there are four nodes in the clustered server. While the RRD-disk requires additional disk read of 29.6GB (7.4GB/node), the RRD-net requires additional 29.6GB network traffics at inter-cluster network. Both methods reduce the service resumption time, but waste system resources to make the backup process ready for service resumption.

4.2 Overhead Analysis

In this subsection, we analyze the overhead imposed by applying the proposed mechanisms to the traditional media servers for service continuity in case of failures. We first define some notations as follows

- N_{node} : the number of nodes in a clustered server
- N_{pproc} : the number of primary processes
- N_{bproc} : the number of backup processes
- T_{diskio} : disk access time for retrieving media contents
- T_{proc} : additional execution time for continuous service in the primary and backup processes
- T_{LSS} : synchronization time for updating status at LSS (refer to Fig. 2)
- $T_{RRD-disk}$: synchronization time for updating status at RRD-disk
- $T_{RRD-net}$: synchronization time for updating status at RRD-net
- ForwdOH : overweight delivery time in inter-cluster network than normal condition due to traffic congestion
- SyncRate : the rate of status synchronization (refer to Fig. 2)

First, as for the case of LSS, there is only the overhead of forwarding status information to the single node. The overhead is expressed in the following equation:

$$OH_{LSS} = \frac{\sum_{N_{pproc}} T_{proc}}{N_{node}} + \sum_{N_{pproc}} T_{LSS} \times ForwdOH \qquad (1)$$

where the first term is trivial since there is no backup processes in LSS and primary processes only send simple status information. The second term is time for delivering status information to the log node. Second, the RRD-disk creates a backup process for each primary process. Each primary process forwards its status to the backup process, which then prepares media contents for reducing the service resumption time. So, we can express the overhead as follows:

$$OH_{RRD-disk} = \frac{\sum_{N_{pproc}+N_{bproc}} T_{proc}}{N_{node}} + \frac{\sum_{N_{pproc}} T_{diskio} \times SyncRate}{N_{node}} + \sum_{N_{pproc}} T_{RRD-disk} \times ForwdOH \qquad (2)$$

where includes the execution time to deliver and receive media contents both at the primary and backup processes. The second term is due to the disk access time for preparing media contents. The last term is due to the delivery of status information to each delivery node. Finally, in the RRD-net, each backup process gets the media contents from the primary process instead of its local disk, resulting in the following overhead equation:

$$OH_{RRD-net} = \frac{\sum_{N_{pproc}+N_{bproc}} T_{proc}}{N_{node}} + \sum_{N_{pproc}} T_{RRD-net} \times ForwdOH \qquad (3)$$

where the second term is due to the delivery of media contents to each delivery node. Note that, as for the synchronization time, the following inequality satisfies

$$T_{RRD-disk} < T_{LSS} < T_{RRD-net} \qquad (4)$$

Let us analyze the above overhead equations. In equation (1), the overhead is directly proportional to the ForwdOH variable that increases linearly with the number of nodes and status synchronization. This is because we assume the single log node. In equation (2), the second term can cause somewhat overload due to heavy disk accesses, but its overhead is inversely proportional to the cluster size. As for the RRD-net, since $T_{RRD-net}$ is larger than T_{LSS} and $T_{RRD-disk}$, the cluster size gives a great influence on the overhead by the second term in equation (3). Through the above equations, we can conclude that, with the same number of concurrent clients, the RRD-disk gets better than the others in terms of the overhead as the cluster size increases. A large cluster size causes a serious overload in both LSS and RRD-net.

5 Simluation Results

This section presents the simulation results for the proposed service continuity mechanisms. The simulation model is implemented using CSIM [14]. Every resource (delivery nodes, disks, inter-cluster network device) in the simulation model is modeled as a single server queue. The model assumes the followings:

- The client requests are uniformly distributed into delivery nodes
- The throughput of inter-cluster network is two times higher than that of disks
- The throughput of inter-cluster network decreases ten percent whenever the number of node becomes double
- Disk throughput decreases in proportion to the number of processes accessing the file system
- A clustered media server concurrently processes 500 clients
- Buffer cache for disk read is not considered

The main goal is to analyze the impact of the proposed mechanisms on the performance of the clustered media server. In order to do this, we ran a set

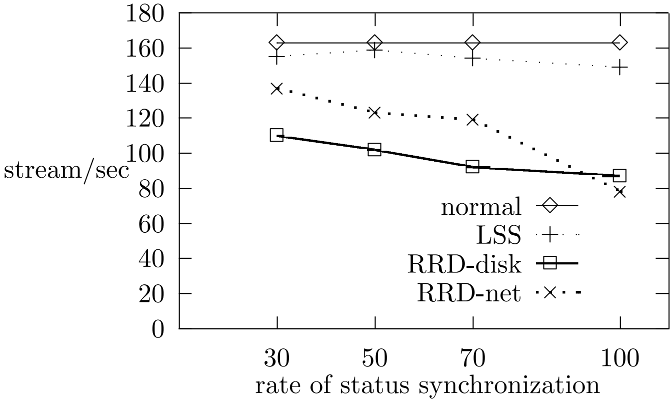


Fig. 3. Media Throughput under Service High Availability (Cluster of 2 nodes)

of simulations with various synchronization rates and cluster sizes. We show the simulation results for two-, four-, and eight-node clusters with various synchronization rates. The x-axis indicates the rate of status synchronization. The higher the number in the x-axis is, the more accurate the consistency is. Let us consider the example (90 minute movies) described in Subsection 4.1. If we keep up 50% status consistency, there are 2,700 synchronizations per a client. The y-axis indicates the number of streams delivered to clients per second.

Fig. 3 shows that the LSS mechanism imposes 2%-9% of performance degradation on the traditional media server, which is indicated with the normal line, when the cluster size is as small as two nodes. Both the RRD-disk and the RRD-net show severe performance degradations because of the heavy disk traffic and network traffic, respectively. Among two RRD schemes, RRD-net demonstrates up to the average 25% performance improvement over RRD-disk except for around 100% synchronization rate. After 70% synchronization rate in RRD-net, the media server throughput is a little decreased due to frequent network traffics. On the other hand, the performance of RRD-disk is smoothly decreased according to synchronization rates.

However, as shown in Fig. 4, as the size of the cluster server increases, the performance of the RRD-disk gets better. This is because the processing power increases to handle disk IO. Compared with two-node cluster, RRD-disk demonstrates up to 25% performance improvement. As for LSS in four-node cluster, the node increase has strong impact on the media server throughput. This shows up to 10% degradation in throughput compared with LSS in two-node cluster. Besides, according to the synchronization rates, the throughput is smoothly decreased, although there is a little change in two-node cluster. In case of RRD-net, it is the similar phenomenon as LSS. However, around 70% synchronization rate, the throughput is more dramatically decreased than before.

The performance gaps between RRD-disk and the other two become bigger as shown in Fig. 5. This is because, as explained in the previous section, the performance overheads of the LLS and the RRD-net increase with the server

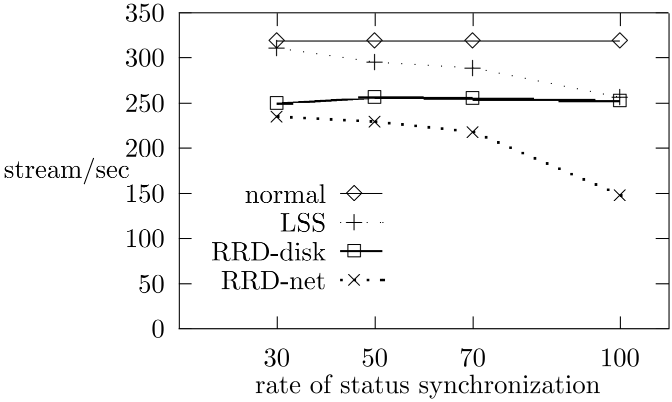


Fig. 4. Media Throughput under Service High Availability (Cluster of 4 nodes)

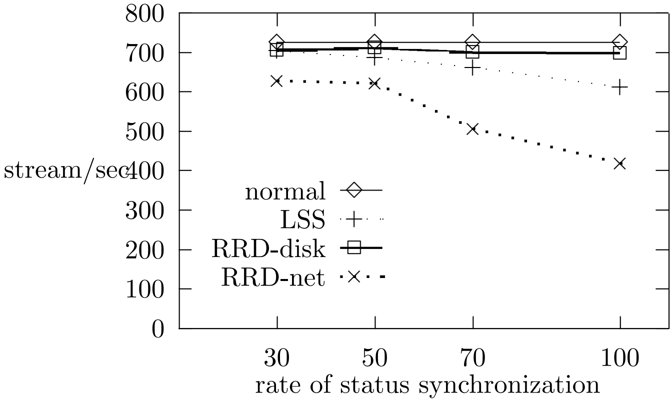


Fig. 5. Media Throughput under Service High Availability (Cluster of 8 nodes)

size, but that of the RRD-disk does not. In all cases, as expected, the performances decrease as the synchronization rates increase. Specially, in case of RRD-net, the symptom of performance degradation happens more early than other mechanisms. The RRD-disk shows a much smoother degradation than other two mechanisms, and shows more up to 18% performance improvement than four-node cluster.

So as to assess the degree of scalability of the proposed mechanisms, we performed another simulation. The results are shown in Fig. 6, which shows the number of streams per node per second, as the cluster size increases from two to eight nodes. This figure shows that the performance of the RRD-disk is getting closer to that of the normal case than those of the LSS and the RRD-net, as the size of the cluster server increases. This means that the RRD-disk is more scalable than LSS and RRD-net mechanisms, although they are also scaled up with cluster size.

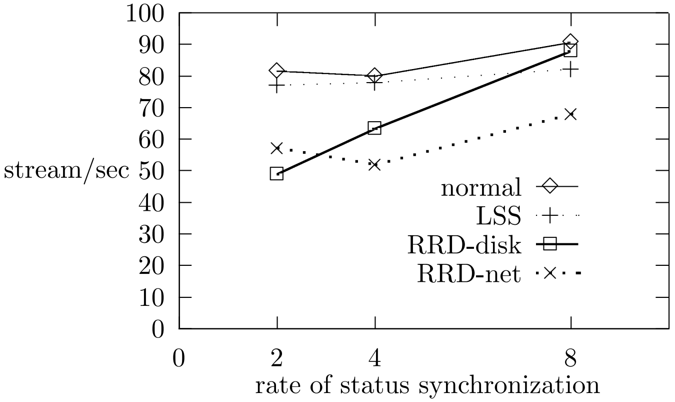


Fig. 6. Scalability

From the above two simulations, we can conclude that the RRD-disk shows much better performance as the size becomes large, and the LSS and RRD-net are useful in small cluster size. Also note that the resumption time of the RRD is much shorter than that of the LSS mechanism.

6 Conclusions

In this paper, we proposed three mechanisms guaranteeing continuous stream services on failures happened during service delivery in wired or mobile Internet. The basic concept for supporting service continuity is the management of status by which we resume the closed services within a short time in the events of failure. Since these mechanisms provide additional functionalities for the traditional media servers, we have to minimize the performance overhead imposed by applying the mechanisms on the traditional media servers. So as to do that, we analyzed and simulated the performance overheads. The simulation results showed that the mechanism overhead was variable according to the cluster sizes and status synchronization rates. Compared with normal throughput provided by traditional media servers not supporting high availability, three mechanisms showed 2%-47% performance degradation. Therefore, we finally knew that selecting a proper mechanism according to the cluster size we could provide continuous stream services for multimedia applications with the minimum overhead. As future works, we are studying the recovery algorithm based on LSS and RRD mechanisms, and also analyzing the cost performance of three mechanisms.

References

1. Craig Freedman and David DeWitt. The SPIFFI scalable video-on-demand server. SIGMOD, (1995)
2. Roger Haskin and Frank L. Stein. A system for delivery of interactive television programming. COMPCON, (1995)

3. Renu Tewari, Dan Dias, Rajat Mukherjee, and Harrick Vin. Real-time issues for clustered multimedia servers. IBM Research Report-RC 20020 (1995)
4. Rajesh Krishnan, Dinesh Venkatesh, and Thmos D.C. Little. A failure and overload tolerance mechanism for continuous media servers. ACM Multimedia, (1997)
5. Renu Tewari, Daniel M. Dias, Rajat Mukherjee, and Harrick M. Vin. High availability in clustered multimedia servers. Proceedings of the 12th International Conference on Data Engineering, (1996)
6. Harrick M. Vin, P.J. Shenoy, and Sriram Rao. Efficient failure recovery in multidisk multimedia servers. FTCS, (1995)
7. S. Berson, S. Ghandeharizadeh, R. Muntz, and X. Ju. Staggered striping in multimedia information systems. Proceedings of the 5th SIGMOD, (1994)
8. Steven Berson, Leana Golubchik, and Richard R. Muntz. Fault tolerant design of multimedia servers. SIGMOD, (1995)
9. F. Tobagi, J. Pang, R. Baird, and M. Gang. Streaming RAID- a disk array management system for video files. ACM Multimedia, (1993)
10. E. Chang and A. Zakhor. Scalable video data placement on parallel disk arrays. Proceedings of Stroage and Retrieval for Image and Video Database II, (1994)
11. Rajat Mukherjee, Daniel M. Dias, Christos A. Polyzois, Jerry Leitherer, and Jeffrey S. Lucash. Cost-performance issues for clustered video servers. IBM Research Report, (1994)
12. Renu Tewari, Rajat Mukherjee, Daniel M. Dias, and Harrick M. Vin. Design and performance tradeoffs in clustered video servers. International Conference on Multimedia Computing and Systems, (1996)
13. Ram Chillarege, S. Biyani and J. Rosenthal. Measurement of Failure Rate in Commercial Software. IBM Research Report RC-19889, (1994)
14. Mesquite Software, Inc. Getting started: CSIM18 simulation engine(C version). (1994)

A SOAP-Based Framework for the Internetworked Distributed Control Systems

Changho Lee, Jaehyun Park, and Yoosung Kim

School of Communication and Information Engineering,
Inha University, Incheon 402-751, Korea
{jhyun,yskim}@inha.ac.kr

Abstract. Emerging IT technologies, specially Internet communication and webbased technologies are adopted to the modern distributed control systems. This paper defines a functional framework for the web-based applications of a distributed control system connected by Internet. XML(eXtensible Markup Language) is used for representing a control system and control devices. These IT technologies make a distributed control system more flexible and scalable than existing distributed control systems with the standard RMI protocols such as CORBA or DCOM.

1 Introduction

Modern Internet and information technologies(IT) enable the control systems to utilize the distributed environment connected by computer networks, such as field-buses, Ethernet, ATM, and wireless networks. By adopting these information technologies in WAN(Wide Area Network), the geometric range for a distributed control system becomes wider and wider, and in turn, a new software framework is required for this new environment [1].

A software for the Internet-based distributed control systems is used to be designed using the standard software components that operate on the top of the industry standard middlewares such as CORBA (Common Object Request Broker Architecture) from OMG (Object Management Group) [2,3] and COM (Component Object Model) from Microsoft [4,5]. Using these kinds of middlewares that support the network-based software components makes a distributed control system more flexible and manageable.

However, because these middlewares primarily aim the general Internet-based applications, they have important problems to be directly used for the distributed control systems [6,7].

- Since two major middleware standards, DCOM and CORBA, are not compatible with each other, it is hard to maintain the interoperability between them [8].
- Since the network messages for these middlewares hardly pass through the security barrier like a firewall, they are not likely used in the distributed control systems across the WAN boundaries.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 195–204, 2002.

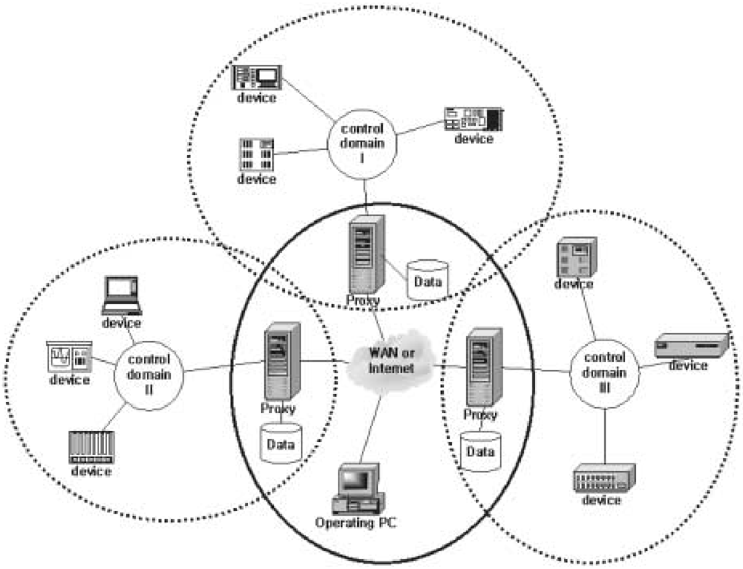


Fig. 1. Global Control Domain

- Since these middlewares are primarily developed for the general-purpose operating systems such as Microsoft Windows or Unix, it is difficult to apply them to the embedded control systems directly [9,10].

To overcome these difficulties in using CORBA or DCOM, a new protocol, SOAP (Simple Object Access Protocol), is proposed as a RMI(Remote Method Invocation) for the Web-based applications. Since SOAP uses HTTP(Hyper Text Transfer Protocol), it is suitable for designing a control systems in WAN environment like Fig. 1. However, the current version of SOAP standard defines only a simple protocol format for a remote method invocation, a SOAP message structure should be defined in detail to use SOAP for a distributed control system. This paper defines the SOAP message structure for a distributed control system for the remote monitoring and control. This paper is organized as follows. Section 2 introduces the background technologies that are needed to design the control system. Section 3 proposes a control domain and SOAP message structure. Section 4 demonstrates the implementation example. Finally, Section 5 concludes this paper.

2 Background Technology

2.1 eXtensible Markup Language (XML)

XML, a standard protocol defined by W3 organization, is a meta-language to describe other languages [11]. Since the underlying philosophy of XML aims flexibility and portability, it is possible to design a platform-independent RMI using

XML. XML can be considered as two kinds of objects: a document and data. XML as a document, is used to make a document and define tags and technologies. XML as a data, is considered as a transfer syntax as well as data types [12]. In addition, hierarchical data structure of XML is very useful to convey the internal data structure between the distributed control systems. Recently, to utilize XML data more efficiently, XML Information Set(XML InfoSet) is proposed. With XML InfoSet, network traffic can be reduced for the realtime communication.

2.2 Simple Object Access Protocol

SOAP (Simple Object Access Protocol) is a HTTP-based RMI(Remote Method Invocation) protocol and has a message structure defined by XML standard [13]. Since only a web server module is necessary to use SOAP, ranges of embedded control systems are able to maintain interoperability with other platforms using SOAP. The packet format of SOAP is similar to that of existing ORPC(Object Remote Procedure Call) that is used in DCOM or CORBA IIOP/GIOP. This similarity makes it easy to convert most of the existing objects and methods written in DCOM or CORBA to SOAP-based objects and methods. SOAP uses URL(Uniform Resource Locator) and URI(Uniform Resource Identifier), for the object references and method requests [14]. Even SOAP has the same function as ORPC, SOAP has following advantages over other RMI methods: First, since SOAP is a plain text, it is easy to bind various protocols. Second, while IIOP(Internet Inter-ORB Protocol) of CORBA and Java RMI requires very bulky infrastructure for a remote object reference or a garbage collection, SOAP doesn't require it.

3 Web-Based Framework for Control Domain

This section defines the SOAP message format and XML document for the application software for a distributed control system.

A large distributed control system, defined as a *control domain*, consists of several sub-domains that include multiple devices as shown in Fig. 1. Internetworking between control domains is gated by a proxy server. The proxy server between control sub-domains gives several merits. First, the proxy server can moderate the network traffic, that is a very important issue for the real-time operation of the control systems. If a real-time control system is open to the public Internet, it experiences high network load as well as attacking from outside. For the real-time operation, the network between control devices should be isolated from the public Internet. This isolation is easily achieved by using the proxy server, with which each control devices exchange their data with each other without being interrupted from the outside traffic. Second, the proxy server can maintain a high level of security. Real-time control system can be efficiently protected from the active attacks such as DOS(Denial of Service). Last but an important point is to compensate the inherit stateless property of

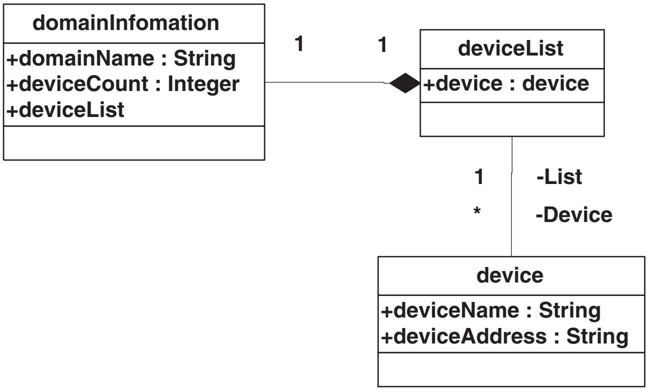


Fig. 2. Domain Description Language for a Control Domain

HTTP. Without any state information, maintaining the control information efficiently is almost impossible and large network load is usually injected. But with a proxy server, state-oriented communication can be efficiently managed even SOAP itself doesn't support it [15].

3.1 Control Domain Abstraction

To use SOAP for a control domain, a control domain itself as well as the services provided by the control devices should be well defined in XML format. These definitions are called Domain Description Language (DDL) and Service Description Language (SDL) respectively, and they reside in the proxy server that represents a particular control domain.

Fig. 2 shows the proposed XML schema of DDL for a control domain. A DDL consists of three major elements: `domainName`, `deviceCount`, and `deviceList`. `domainName` represents the name of current control domain. `deviceCount` stands for the number of devices registered in a control domain. The devices within a control domain is described in the `deviceList` that has two elements: `deviceName` and `deviceAddress`. With `deviceName` element, user can fetch its SDL (deviceName.sdl) from a proxy server, and, in turn, get services and parameters. A `deviceAddress` element is used to inform the proxy server of the physical address of a device. In Internet, `deviceAddress` is represented by IP Address or the hostname.

3.2 SOAP Message for Control Domain

This section proposes a SOAP message format for a control domain whose abstraction is defined in the previous section. Since current SOAP standard defines only the minimal set of SOAP header, the detail structure of header and body for a specific domain should be defined. Fig. 3 shows the SOAP message structure for the control domain proposed in this paper.

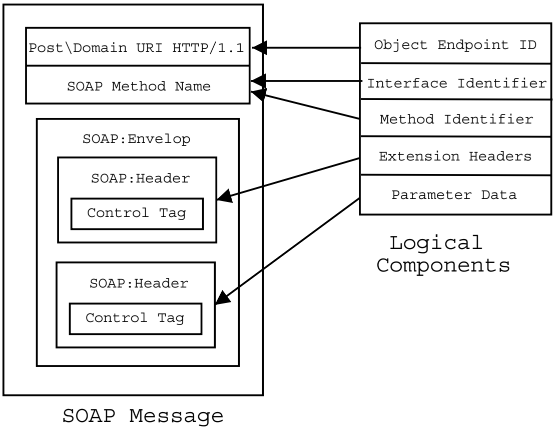


Fig. 3. SOAP Message Format

```
<!ELEMENT SOAP-ENV:Envelope (SOAP-ENV:Header)>
<!ATTLIST SOAP-ENV:Envelope xmlns:xsi   CDATA #REQUIRED
                            xmlns:SOAP-ENV  CDATA #REQUIRED >
<!ELEMENT SOAP-ENV:Header   (trans:transfer , const:constraint )>
<!ATTLIST SOAP-ENV:Header   xsi:type CDATA #REQUIRED >
<!ELEMENT trans:transfer    (trans:to , trans:from )>
<!ATTLIST trans:transfer    xmlns:trans CDATA #REQUIRED
                            xmlns:agr   CDATA #REQUIRED
                            SOAP-ENV:mustUnderstand CDATA #REQUIRED>
<!ELEMENT trans:to (trans:address )>
<!ELEMENT trans:address EMPTY>
<!ATTLIST trans:address xsi:type CDATA #IMPLIED >
<!ELEMENT trans:from (trans:address )>
<!ELEMENT const:constraint  (const:timestamp ,  const:timeout  )>
<!ATTLIST const:properties  xmlns:const CDATA #REQUIRED
                            SOAP-ENV:mustUnderstand CDATA #REQUIRED>
<!ELEMENT const:timestamp   (#PCDATA )>
<!ELEMENT const:timeout     (#PCDATA )>
```

Fig. 4. SOAP Control Tags

Control Tags. Fig. 4 shows the internal structure of SOAP header. As described above, the control domain defined in this paper uses a proxy server, all information related to message transfer should be registered in the proxy server. This information includes address information and timing constraint required for the realtime control. To provide information of control devices for the proxy server, *control tag* is defined in SOAP header, that includes two tags.

`transfer tag`

In the control domain proposed in this paper, since control devices are isolated from the external domains as described above, user should provide all of the information to drive a device in order that a proxy server could interface with control devices instead of clients. For this purpose, a mandatory element, `transfer tag`, is de- fined. `transfer tag` includes the location information of SOAP message source and destination.

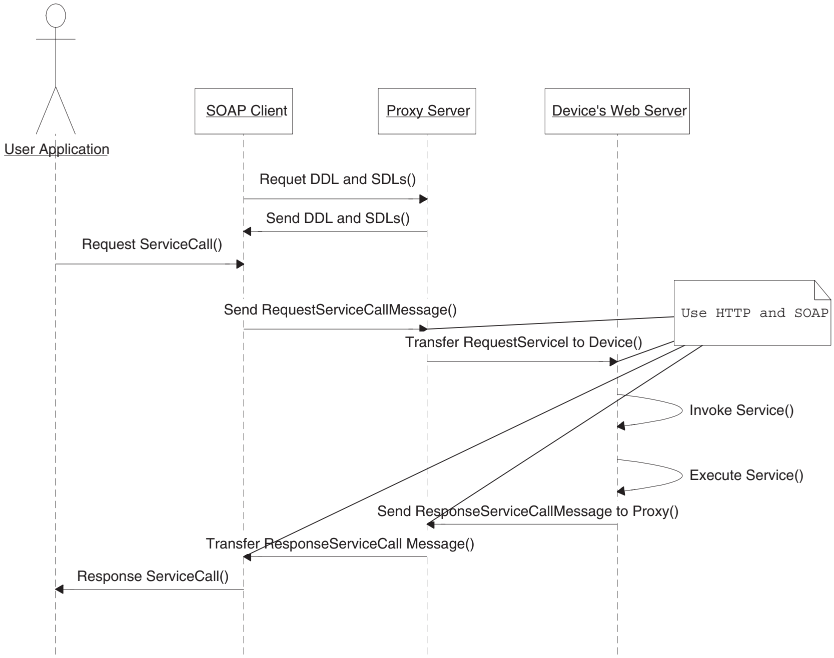


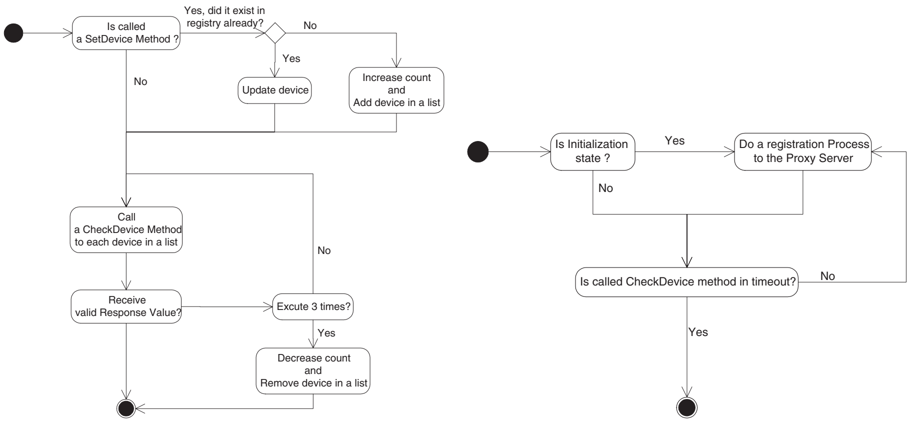
Fig. 5. SOAP-Compliant Processing

`constraint tag`
A constraint tag contains the timing information to decide whether a tranferred message meets the real-time constraints or not. Sender records the `timestamp` in a timestamp element, and timeout value in a `timeout` element. If the timeout value is expired, a message is considered as an invalid message.

Service Call. For the security reason, a user is required to be authorized for accessing the control system. A proxy server can authorize user to access the control domain using an industry standard authentication mechanism. After authentication process, users can obtain the domain information written in DDL and SDL from the proxy server. With the information obtained, the service request and the response messages are exchanged as shown in Fig. 5.

3.3 Registration and Management of Control Devices

Since DDL locates in the proxy server, each device must register its services to the proxy server. A control device can register its IP address and SDL to the proxy server through a web service defined in Fig. 7. As shown in Fig. 6(a), when the proxy server receives a registration message from a device, it determines whether a device is already registered in the `deviceList` or not. If a device is not



(a) Serverside

(b) Clientside

Fig. 6. Process State Diagram

```
ProtoType : boolean SetRegister(string, string, string); int CheckDevice(int);

<ss:schema  id="RegisterStuff" targetNamespace="http://server-url/services.xml"
        xmlns:dt="http://www.w3.org/1999/XMLSchema" xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="SetRegisterReq"/>
        <type>
            <element name="Addr" type="dt:string" />
            <element name="DomainName" type="dt:string" />
            <element name="SDL" type="dt:string" />
        </type>
    <element name="SetRegisterRes">
        <type>
            <element name="RetureCode" type="dt:boolean" />
        </type>
    </element>
</ss:schema>

<ss1:schema  id="HeartbeatStuff" targetNamespace="http://server-url/services.xml"
        xmlns:dt="http://www.w3.org/1999/XMLSchema" xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="CheckDeviceReq"/>
        <type>
            <element name="ConnectionReference" type="dt:int" />
        </type>
    <element name="CheckDeviceRes">
        <type>
            <element name="ReferenceValue" type="dt:int" />
        </type>
    </element>
</ss1:schema>
```

Fig. 7. SDL of Registration and Management Methods

registered in the list, the proxy server adds a new device into the list. Otherwise, it just updates the information of the device in the list. After the registration process completes, the proxy server uses the `CheckDevice` method to check the device is removed from a domain. If the device returns failure code or doesn't

Table 1. Basic Services

Basic Device	Basic Readable	Basic Writable	Basic Event
Start()	ReadData()	WriteData()	SetEvent()
Stop()	SetRxTimeOut()	SetTxTimeOut()	ClearEvent()
Pause()		SetLock()	
Shutdown()		SetUnLock()	

respond to the `CheckDevice` message, the proxy server assumes that it is not in a domain and removes it from the list. On the contrary, if there is no `CheckDevice` message from the proxy server during a certain period, a control device requests a registration process as shown in Fig. 6(b).

3.4 Basic Control Services

To adopt the SOAP protocol to the control systems, basic control services should be defined. Table 1 shows four service categories: `BasicDevice`, `BasicReadable`, `BasicWritable,` and `BasicEvent`. `BasicDevice` includes the fundamental services such as device initialization and operation control. `BasicReadable` and `BasicWritable` provide the services neccessary for reading and writing data from/to the control systems. Last service group, `BasicEvent`, includes the services used for the event handling that plays an important role for the control system monitoring. The interface information of the basic control services written in SDL(Service Description Language) is exported to the exteranl users. To use these services within a control domain, the procedural sequence is shown in the sequence diagram of Figure 8.

4 Implementation

Using the Web-based frameworks proposed in this paper, a demonstration system is implemented. Target application domain is a distributed controller for a semiconductor manufacturing machine, Samsung's commercial chip mounter (Samsung CP-40). To demonstrate the O/S-independent characteristics of the protocol, the devices in the domain are designed on two different operating systems: Microsoft Windows and embedded Linux.

SEMI (Semiconductor Equipment and Material International Inc.) introduced the SECS (SEMI Equipment Communication Standard) protocol to reduce the cost and to improve communication ability between hosts and equipments. This protocol consists of SECS-I for the message communication and SECS-II for the message format. SECSI uses point-to-point protocol such as RS-232. Recently, HSMS (High-Speed SECS Service) based on TCP/IP is used widely. GEM(Generic Model for Communications and Control of SEMI Equipment) is the protocol that is used between semiconductor equipment and host computer on the top of SECS-I, II and HSMS. In this paper, the proposed SOAP message is used to integrate GEM-based semiconductor manufacturing

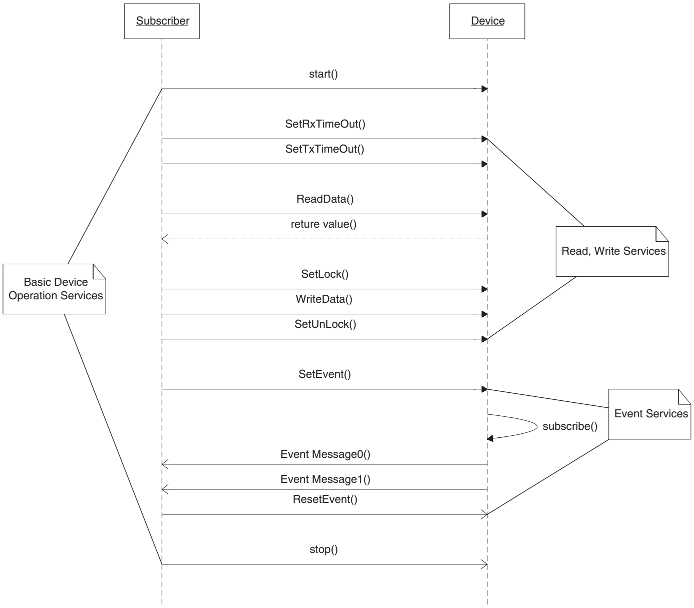


Fig. 8. Service Sequence Diagram

machines. Implemented system consists of a proxy server and several devices. The proxy server is implemented on a Microsoft Windows 2000 platform and IIS5.0. It stores and supplies the device's information in DDL and exchange various SOAP messages. As an authentication protocol, the Kerberos authentication protocol is used, that is also independent of platforms. Various target control devices are implemented with Microsoft Windows family(2000, CE) and embedded Linux.

5 Conclusions

Emerging IT technology, specially Internet and web-based communication, is likely to be adopted to distributed control systems recently. This paper defines web-based control domain connected by Internet, and proposes a functional framework for it. Control domains defined in this paper are connected with each other via a proxy server. Under the control of proxy server, the real-time operation and the security of control domain can be maintained systematically. For openness, a control domain and control devices are abstracted in XML, and SOAP is used for RMI(remote method invocation). Based on SOAP, a set of services for the control and monitoring operation are defined. With adopting these IT technologies, a distributed control system becomes more flexible and scalable than existing distributed systems even they use a standard RMI method such as CORBA or DCOM. At last, the proposed protocols can be easily implemented on the control devices regardless of operating systems, total costs and developing time can be reduced. One drawback of the proposed protocol is high overhead caused by raw XML data and HTTP protocols. These problems can be, however, reduced with XML Infoset mechanism and other methods.

Acknowlegements

The work reported in this paper is partially supported by Inha University Research Grant(Grant number 20695).

References

1. Lee, W., Park, J.: A design of infrastructure for control/monitoring system in the distributed computing environment. In: Procceedings of 15th Korea Automation Control Conference, Yongin, Korea (2000)
2. OMG: Corba services:common object services specification. Technical report, OMG,Inc (1996)
3. Orfali, R., Harkey, D., Edwards, J.: Instant CORBA. John Wiley & Sons (1998)
4. Microsoft: DCOM technology overview. Technical report, Microsoft Corporation (1996)
5. Caron, R.: Web services and the soap for visual studio 6.0. MSDN Magazine **15** (2000) 62–73
6. Grimes, R.: Professional DCOM Programming. Wrox (1997)
7. Daniel, J., Vallee, B.T.V.: Active COM : An inter-working framework for CORBA and DCOM. Proceedings of the International Symposium, Distributed Objects and Applications (1999)
8. Chang, E., Annal, D., Grunta, F.: A large scale distributed object architecture - CORBA and COM for real time systems. Third IEEE International Symposium (2000) 338–341
9. Chen, D., Mok, A., Nixon, M.: Real-time support in COM. In: Proceedings of the 32nd Hawaii International Conference on Systems Sciences, Hawaii, USA (1999)
10. Chisholm, A.: OPC Overview. Technical report, OPC Foundation (1997)
11. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E.: Extensible markup language (XML) 1.0 (second edition). Technical report, W3C (2000)
12. Box, D.: Essential XML beyond Markup. Addison Wesley (2000)
13. Sturm, J.: Developing XML Solutions. Microsoft Press (2000)
14. Box, D.: HTTP + XML=SOAP. MSDN Magazine **15** (2000) 67–81
15. Microsoft: Universal plug and play device architecture. Technical report, Microsoft Corporation (2000)

An Efficient On-Line Monitoring BIST for Remote Service System

Sangmin Bae, Dongsup Song, Jihye Kim, and Sungho Kang

Dept. of Electrical and Electronic Engineering, Yonsei University
134 Shinchon-Dong, Seodaemoon-Gu,
120-749 Seoul, Korea
shkang@yonsei.ac.kr

Abstract. Home networking has been developing rapidly due to the increase of the internet users and the advent of digital economy. In this context, the quality and service under guarantee for internet intelligence electric home appliances has become quite important. Therefore, to guarantee the performance of the appliances, on-line testing for detecting latency faults should be performed. In this paper, we develop a new efficient architecture of on-line BIST. In addition we propose a remote service system for CSM (Customer Satisfaction Management) based on home networking.

1 Introduction

Recently, the worldwide population of internet users is increasing rapidly, which leads to new trends such as the extension of supplying multi-PC for domestic use and the fusion of telecommunication and electric home appliances, etc. Hence a need for Internet intelligence electric home appliances is rising and internet application field is expanding from computers to electric home appliances. Therefore, home networking industry has developed recently more than 15% in the worldwide market, which is causing tremendous far-reaching effects on economy. Now, home networking is an important issue in the information technology field[1][2]. Home networking provides home electric appliances which have more intelligent functions and higher performance. However, consumers demand more reliable appliances and more convenient service under guarantee. Therefore rigorous testing becomes more important to assure the quality and service of home appliances. There are two kinds of testing, off-line testing and on-line testing. The former means that the testing is performed while the system is not operating, and the latter means that the testing is performed during system operation. Usually off-line testing is used to improve the quality by not shipping products which have defects. However using on-line testing, the service costs which include parts and labors can be reduced. In addition, the improvement of service quality can greatly encourage purchasing power of the customers[3]. In this paper, we develop a new efficient architecture of on-line BIST (Built-In Self Test) for identifying faults during system operation. This can be used in a remote service system based on home networking.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 205–213, 2002.

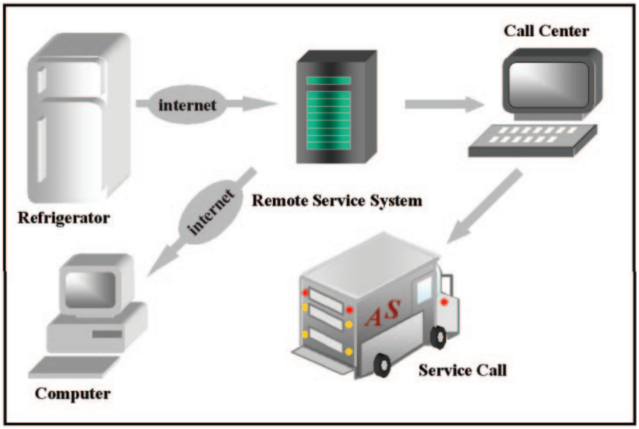


Fig. 1. Example of the remote service system

2 Remote Service System

To construct an efficient remote service system, several things should be considered. Firstly DFT(Design for Testability) should be considered from the beginning. We can find errors of the designs and follow the systematic designs efficiently using DFT. Secondly, suitable test processes should be followed. There are two kinds of test processes: off-line testing and on-line testing. Off-line testing detects the faults caused during the manufacturing process. On the other hand on-line testing detects the faults caused by defective operations and bad effects from the outside during operation. To apply off-line testing, BIST and/or tester can be used. On the other hand, on-line testing must use BIST. Thirdly, the network which connects all systems should be formed. Finally, the databases obtained from the analysis of the test information should be prepared[4][5]. The system constructed with the above design, has many advantages. Especially with on-line BIST, not only the latency faults that can't be detected by off-line testing are found but also the service is improved by repairing the faults detected by on-line BIST. Suppose that there is a defect in a refrigerator. It is detected by on-line BIST before the consumer finds out. And this information will be sent to the manufacturing company via internet and the message is sent to the user via e-mail or SMS (Short Message Service) at the same time. Next, the company analyzes the problem. Then the process of repairing or exchanging parts is followed. All these steps are shown in Fig. 1. Consequently, the concept of the "intelligent service under guarantee" can be constructed based on the remote service system. This concept means that the service centers provide maintenance service for home appliances before the consumers ask for it. This system has a number of advantages: the cost of service can be reduced, high quality of service can be provided and the quality of products can be improved.

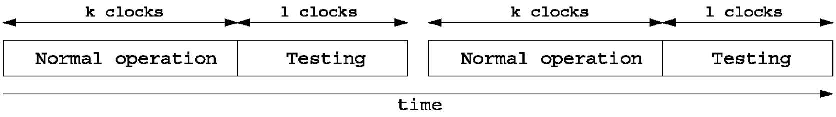


Fig. 2. Overlapping of testing and normal operation on the CUT

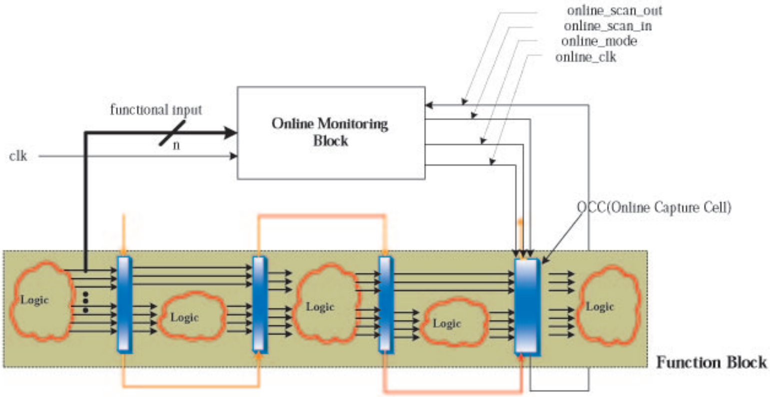


Fig. 3. Overview of on-line monitoring system

3 Effective On-Line Monitoring Architecture

3.1 Overview of On-Line Monitoring System

There are two kinds of on-line testing. One is inserting additional testing modes and the other is on-line monitoring [6]. Inserting additional testing modes between normal operation actions is illustrated in Fig. 2. This method has a merit that it could apply deterministic test patterns which have high fault coverage through additional test modes. But it is difficult to determine when the test modes are inserted. In other words, it is difficult to estimate when system blocks enter into the idle time from normal operation and how long the idle time is continued. Therefore, it is not easy to use this method. On the contrary, on-line monitoring BIST determines whether the system has faults or not by comparing a monitored event come out from the functional input with an estimated operating result of the fault free system. For example, let's look at on-line monitoring pipelined function block shown in Fig. 3. In this case, on-line monitoring circuit has a trait that a signal appears at the input terminal and after four clocks the specific signal combination appears at the output terminal. Therefore in a fault free situation, it is possible to predict an output signal combination after four clocks by monitoring the input signal combination. OMB(On-line Monitoring Block) in Fig. 3 detects a monitored signal combination and captured signal after the estimated number of system clocks. And by observing this captured signal, the defects of the on-line system are detected. This paper suggests an effective test structure and a controlling method at the on-line state.

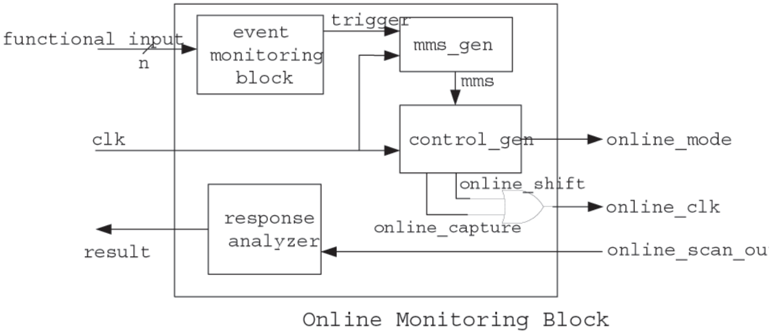


Fig. 4. On-line monitoring block

3.2 OMB (On-Line Monitoring Block)

OMB(On-line Monitoring Block) examines a input signal combination on the function block and performs a corresponding test action when the predetermined signal combination happens. OMB shown in Fig. 4 regards the functional inputs and the system clock of the function block as inputs. Event Monitoring Block(EMB) observes the occurrence of the pre-determined functional input combination. We call this input combination as an ‘event’. As a result, EMB monitors the functional input combination, enables the trigger signal and has the on-line monitoring action start when the ‘event’ occurs. mms_gen is a block that generates a signal required for state changing of control_gen. control_gen is a finite state machine consisted of 4 states. It generates a control signal required by the OCC(on-line capture cell). Response Analyzer compares the signal captured by on-line monitoring actions to the signal expected in a fault free case.

Event Monitoring Block. An event of EMB is selected in consideration of the function block’s feature, because the number of monitoring event determines the size of the hardware. So the input combination used frequently by the function blocks or critical to system operations is regarded as an event of EMB. At this time, there is a trade-off between the credibility of on-line operations of a system and the overhead of the hardware. That is, the more the number of events, the more credible is a system. As a result, the overhead of the hardware would increase.

control_gen. The control_gen block is an FSM made up of 4 states and it changes its conditions by signal of the mms_gen block and generates the control signal on OCC. Fig. 5 is a state diagram of control_gen. control_gen is made up of four states: IDLE, MOVE, SHIFT, and CAPTURE. Each number (0 and 1) beside states means the value of mms in clk’s negative edge.

– IDLE state: IDLE state means a situation before an event is detected in monitoring block. This state is maintained by mms, which is 0 and does not generate capture or shift actions by the control signal on OCC. In this state, the function block operates only functional actions.

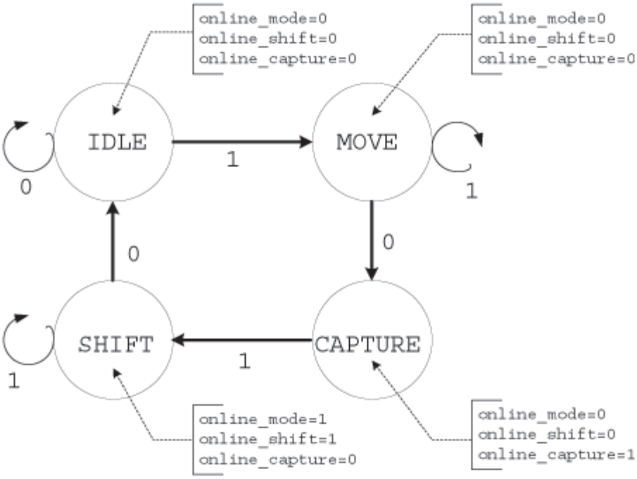


Fig. 5. State diagram of control_gen

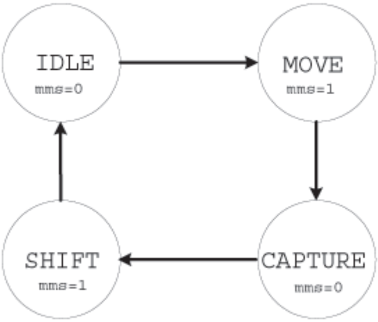
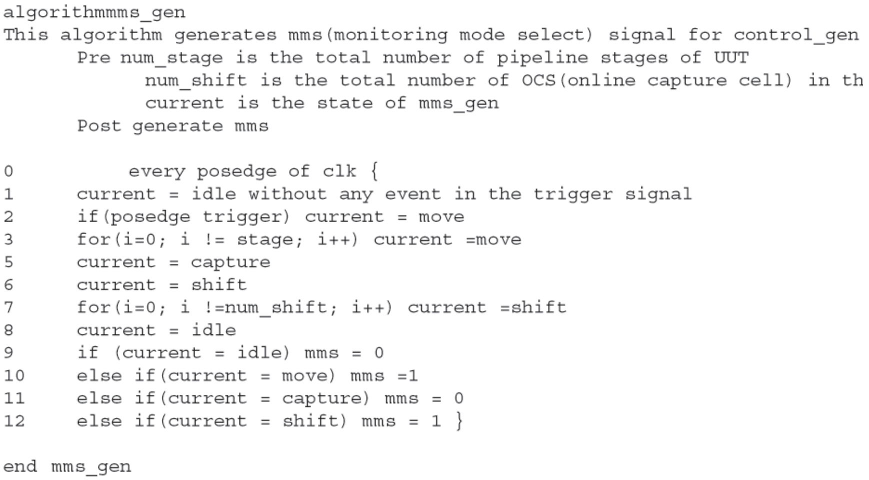


Fig. 6. State diagram of mms_gen

– MOVE state: During the process of detecting an event and transferring its result to the output, control_gen is in MOVE state. For example, like Fig. 3, it takes 4 clks for the expected result to come out at the output. So control_gen is allowed to maintain MOVE state during an event propagation at the output of the function block.
– CAPTURE state: A detected event was propagated to the output of the function block and captured by the OCC.
– SHIFT state: In SHIFT state, the output captured in CAPTURE state is moved to the Response Analyzer. The value of mms in SHIFT state maintains 1.

mms_gen. mms_gen is an FSM, which has 4 states and generates mms required to change a state of control_gen. Fig. 6 shows a state diagram of mms_gen and Fig. 7 shows an algorithm of mms_gen. mms_gen includes a num_stage register, which has information about the number of pipeline stages in the function block and num_shift register, which has information about the number of OCCs. These two registers are hardwired registers.



```
algorithmmms_gen
This algorithm generates mms(monitoring mode select) signal for control_gen
      Pre num_stage is the total number of pipeline stages of UUT
           num_shift is the total number of OCS(online capture cell) in th
           current is the state of mms_gen
      Post generate mms

          every posedge of clk {
      current = idle without any event in the trigger signal
      if(posedge trigger) current = move
      for(i=0; i != stage; i++) current =move
      current = capture
      current = shift
      for(i=0; i !=num_shift; i++) current =shift
      current = idle
      if (current = idle) mms = 0
      else if(current = move) mms =1
      else if(current = capture) mms = 0
      else if(current = shift) mms = 1 }

end mms_gen
```

Fig. 7. Generation algorithm of mms signal

Response Analyzer. A captured result is passed to the compression unit named Response Analyzer instead of being transferred to the external circuit directly, and a signature with a very short length is generated. And then, the signature is directly transferred to the external circuit. By doing so, we can detect malfunctions without comparing the responses of all the test patterns one by one.

3.3 OCCs (On-Line Capture Cells)

OCCs that form the last pipeline stage of the function block obtain the result for a event which is passing through a circuit. In addition, the shift operation of OCCs takes place for the captured value to be observed in Response Analyzer. Fig. 8 shows OCCs. As the latest ICs are more complicated, testing also becomes more difficult to perform. Unless it is considered in the early design state, testing ICs might be almost impossible. Scan is a technique which replaces all flip-flops to scan cells to improve the controllability and the observability of the circuit. An OCC is a modified scan cell by adding MUX2 and a capture cell for on-line monitoring. In a state where an event is not activated, it is the same as the operation of the standard scan cell because a positive edge by the control_gen is not generated at online_clk signal. But if an event is generated in the function block, control_gen is moved to CAPTURE state via MOVE state and online_clk makes the functional output captured in capture cell. Then the captured result moves to the Response Analyzer through the serial path in every online_clk's positive edge generated in control_gen's SHIFT state.

4 Results

In this section we model the proposed on-line monitoring architecture in HDL and perform functional verification. Fig. 9 shows an example of a function block

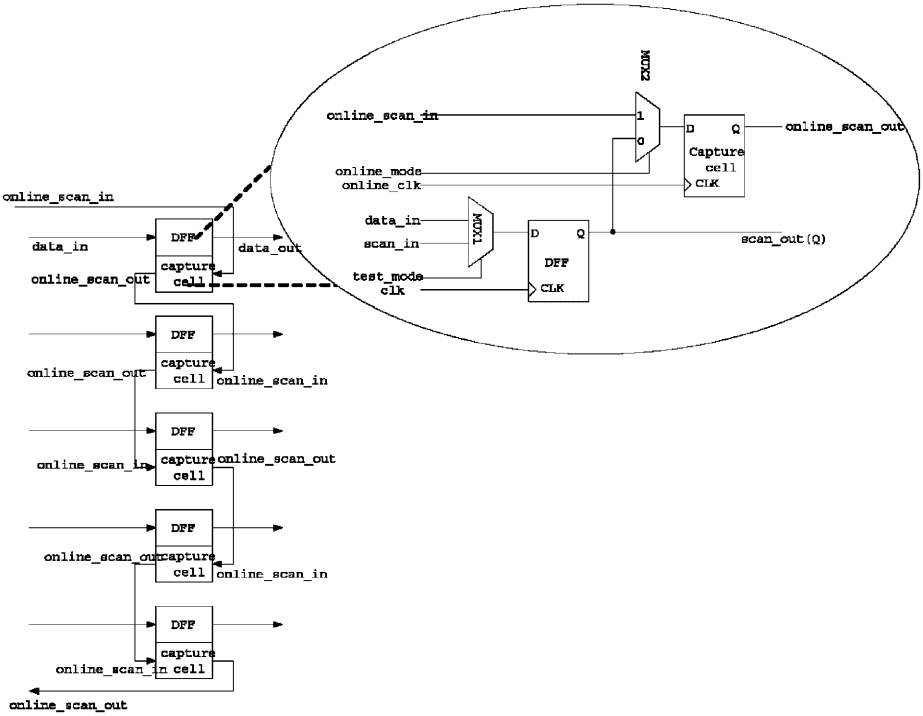


Fig. 8. OCC(On-line Capture Cell)

which has a latency fault. Assume that the latency fault is not detected during off-line testing and it can be modeled as a stuck-at-0 fault. Also, suppose that in the whole system the input combination of 11101 and 01010 occurs frequently according to the function of the function block. The on-line monitoring block may capture the events of 11101 and 01010 by monitoring of inputs and then it can be found out whether the function block operates normally or not. The trigger is enabled when the func_vectors are 11101(1D) or 01010(0A). After the trigger is enabled, the on-line monitoring block stays on the move state until the output signal is visible at the outputs. The response analyzer compares the expected results and the captured results. In Fig. 10, the on-line captured result of the first event is 00010(02), and the result of the second is 10100(14). If the system during on-line testing is fault free, the expected results are 00010(02) and 10101(15) respectively. The simulation shows that the captured result in the second event differs from the expected result due to the stuck-at-0 fault. So the P/F indicates whether the latency fault occurs or not. Since there have been only a few researches about on-line monitoring, it is very difficult to compare the new work with the previous works. The simulation results prove that if a function block has latency faults, the proposed on-line monitoring architecture can detect its malfunction correctly. This can be used as a remote control system.

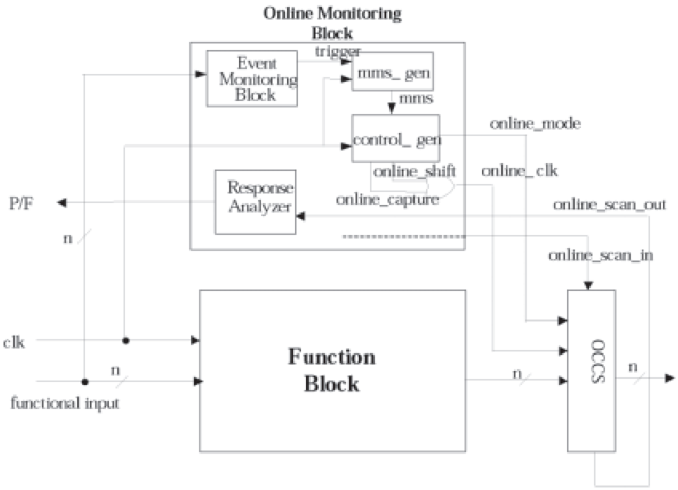


Fig. 9. Example function block

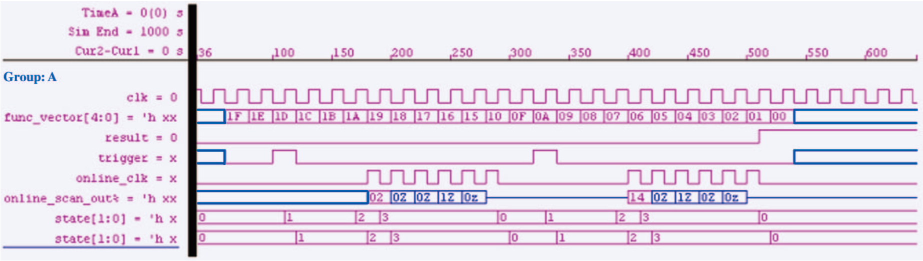


Fig. 10. Simulation result

5 Conclusion

In this paper, we have proposed a new on-line BIST architecture to detect latency faults that can happen during the application of the system chip. The proposed on-line BIST architecture establishes the functional vectors that are effective for testing its operational fault as an event. The on-line BIST monitors whether the established event occurs or not, so when the event occurs, it captures the response of the UUT(Unit Under Test) during on-line testing. It also operates on-line testing that examines whether latency faults occur or not by comparing the expected outputs with the captured outputs. The new architecture minimizes the affects to the system operation due to on-line testing. The on-line monitoring technique can be applicable in many ways. Most people expect that service network will change from the existing service under guarantee system to the intelligent service under guarantee system because of the appearance of household electric goods in web application lately. Therefore the new proposed on-line BIST through on-line monitoring is very promising and has various applications.

References

1. AGRAWAL, S.K., etc: Resource based service provisioning in differentiated service networks. Proc. Of IEEE International Conference on Communications (2001) 1765–1771
2. HUANG, K.L., etc: web-based product design review: implementation perspective. Proc. of International Conference on Computer Supported Cooperative Work in Design (2001) 261 – 266
3. MARSHLL, P., etc: Home networking: a TV perspective. Electronics & Communication Engineering Journal (2001) 209 –212
4. SALZMANN, C., etc: Design for test and the cost of quality. Proc. Of International Test Conference (1988) 302–307
5. BRAUNE, D., etc: ASIC design for testability. Proc. Of ASIC Seminar and Exhibit (1989) 1–10
6. AL-ASAAD, H., etc: On-line built-in self-test for operational faults. Proc. Of AUTOTESTCON Proceedings (2000) 168–174

TCP Performance Enhancement Using FG–LIW (Fairness Guaranteed – Larger Initial Window) Scheme in the Wired–cum–Wireless Network Environment

Inho Roh[1] and Youngyong Kim[1]

Dept. of Electrical and Electronics Engineering, Yonsei University, Shinchon-Dong
Seodaemoon-ku, Seoul 120-749, Korea
{ihroh,y2k}@yonsei.ac.kr

Abstract. TCP(Transmission Control Protocol) is the prevalent reliable transport protocol used for the most popular Internet services like web browsing and e-mail. Since these services are the dominant applications on the Internet, TCP controls the majority of today's Internet traffic. These services usually involve transmission of relatively small amounts of data [1]. In other words, when TCP connections are opened for the data transfer, there exists large probability that the whole transfer is completed while the TCP sender is still in the slow start phase. Therefore, the TCP connection never manages to fully utilize the available bandwidth. Motivated on this fact, we proposes a small change to TCP that may be beneficial to short-lived TCP connections and those over links with large RTT(Round Trip Time)s. Our proposed scheme can save several RTTs during the initial slow-start phase and maximize the wireless bandwidth utilization guaranteeing fairness without degradation of other traffics. Simulation shows significantly improved performance in TCP throughput.

1 Introduction

The TCP provides a reliable, connection-oriented, bi-directional stream between two applications on the Internet [7]. TCP controls flow so that the sender does not send faster than the receiver is prepared to receive. In addition, TCP provides congestion control so that the sender avoids sending fast enough to cause network congestion and adapts to any congestion it detects by sending in slower speed. These days, most implementations of TCP are using gradual increase in congestion window in order to avoid congestion and in order to guarantee fairness. Even though these characteristics may give us stable performance for long TCP flows, it can result in under-utilized performance for the short-lived flow. In this paper, we propose an attractive solution to remove delayed ACK penalty and to maximize the network utilization without degradation of other traffics in the wired-cum-wireless environments.

This paper is organized as follows. Section 2 explains an assumption used in this paper. Section 3 reviews TCP short flow analysis and advantage of

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 215–221, 2002.

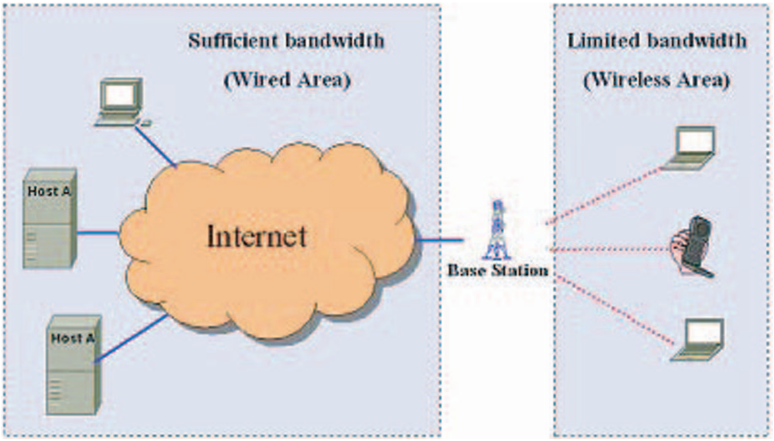


Fig. 1. Wired-cum-wireless Environment

LIW(Larger Initial Window). In section 4, we describe our proposed scheme. Section 5 shows our improved performance through simulation result, and section 6 summarizes our conclusions.

2 Assumption

In wired-cum-wireless network environment, current high speed link technologies like optic fiber makes it possible to transmit bursty traffic over wired link part without loss while wireless link part offers limited data rate. Current memory technology also makes it possible to provide us with high system capacity and the price of system memory is getting cheaper and cheaper. Therefore, most TCP end systems could have enough buffers (bigger advertisement window) to accommodate a moderately bursty traffic of larger initial window of TCP sender without loss. This can also be applied to router to have drop tail queue. Considering these trends, we can assume that future network congestion condition and bottleneck occur mostly in wireless part and the congestion of wired part can be neglected. Another assumption is that BS(Base Station) can snoop TCP messages in 3 way handshake stage for TCP connection like snoop protocol [2].

3 Short TCP Flows Analysis and Larger Initial Window

3.1 Short TCP Flows Analysis

According to [1], for the optimal short TCP flows (assuming no loss), the time required to open k simultaneous TCP connections and to transfer a total of data bytes, incorporating the time for the handshake, delayed acknowledgement, and slow start from an initial window of w segments, is

$$t = log_r(\frac{data(r-1)}{w \cdot MSS \cdot k} + 1) \cdot RTT + RTT + t_{delack} \quad (1)$$

Ratio r, where $1 < r \leq 2$, is an increasing rate of cwnd(congestion window) compared to previous cwnd according to acknowledgement strategy. If TCP is implemented with acknowledging every segment, ratio r is (1+1/1)=2 and with delayed acknowledgement r is (1+1/2)=1.5. From the equation (1), we can characterize the performance of TCP as a function of RTT(Round Trip Time), MSS(Maximum Segment Size), initial window size, delayed ACK policy etc. In standard TCP, these factors are k=1 with a single connection, MSS=1460 bytes, r=1.5 with delayed ACKs and w=1 with one segment initial window.

We can save transfer time and improve performance for short flow by controlling these factors. Among these factors, RTT cannot be controlled in TCP endpoint. In case of MSS, bigger MSS may be supported in very few paths. Even though it is supported in all paths, bigger MSS may cause bigger size of retransmission for the small bit error. In wireless environment which is more prone to transmission losses [8], bigger MSS is not good solution. Simultaneous TCP connections of k also can save transfer time. However, this solution is inadequate since it should maintain multi-connections and should take an additional connection overhead. Delayed Ack penalty (t_{delack}) is also deteriorating factor and should be considered to improve performance.

Therefore, it is very natural idea to control r and w in order to improve TCP performance like DAASS(Delayed Acknowledgement After Slow Start) [3] and LIW(Larger Initial Window) [5]. Without analytical derivation of equation (1), one can intuitively guess that LIW would provide a good performance. However, LIW may be unable to guarantee fairness with other TCP traffics. Thus, fairness must be considered to use LIW scheme.

3.2 Advantage of Larger Initial Window

LIW(Larger Initial Window)'s advantages is to save transmission time [5]. For many email and web page transfers that are less 4K bytes, the LIW would reduce the data transfer time to a single RTT(Round Trip Time). LIW can also eliminates delayed ACK timeout during the initial slow-start phase for TCP operation employing delayed ACKs [4]. Even though delayed ACK timeout does not happen as sender sets enough bigger RTO(Retransmission Timeout) than delayed ACK time, LIW can eliminate at least delayed ACK penalty of about 100ms that may cause a significant degradation for TCP short flows. LIW also would be of benefit for high bandwidth, large propagation delay TCP connections, such as those over satellite links. Nevertheless the latest RFC [4] defining TCP's congestion control recommends a gradual increase in IW specifying that "the initial value of congestion window MUST be less than or equal to 2*SMSS bytes and MUST NOT be more than 2 segments" since LIW(Larger Initial Window) can cause another traffic's degradation under the condition of competition,. Only if fairness can be guaranteed, LIW will be a very good solution to improve TCP performance.

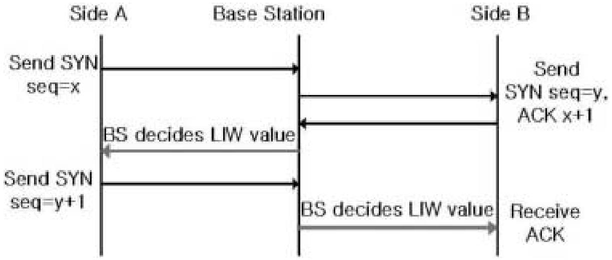


Fig. 2. Three-way-handshake with FG-LIW

4 FG-LIW (Fairness Guaranteed-Larger Initial Window)

4.1 Fairness Guaranteed LIW

Before sending any data, the two TCP endpoints must establish a connection between themselves by a three-way-handshake. To guarantee fairness, FG-LIW value is decided in BS(Base Station) according to remaining available bandwidth during the connection setup. It is delivered to both communication parties through 2-bit of 6-bit reserved field of the TCP header during the three-way-handshake (Figure 2). The first segment of a handshake can be identified because it has the SYN bit set in the code field [7]. TCP can be initiated by wired endpoint as well as wireless endpoint and can be used as duplex line. Thus mobile host can use established connection not only as uplink but also as downlink.

4.2 FG-LIW Decision Function of Base Station

BS measures available bandwidth and calculates FG-LIW value. Available bandwidth can be measured at every TCP connection setup, based on (2). We denote Maximum.BW as the maximum bandwidth BS can handle, γ as channel fading factor, Current.Used.BW as the bandwidth other TCP connections are using currently.

$$Available.BW = \gamma * Maximum.BW - Current.Used.BW \tag{2}$$

Channel fading factor, γ, where $0 \leq \gamma \leq 1$, is time varying factor according to current wireless channel condition. Choosing a value for γ close to 1 means that air condition is good and there is less channel fading degradation. Initial window size is determined in the range of 1 to 4 using 3-level-threshold (Figure 3). IW value bigger than 4 does not recommended because it can cause too much network aggressiveness and excessive packet drop rate [5]. If the available bandwidth is greater than Low-threshold, FG-LIW will give us better performance than traditional TCP operation guaranteeing fairness. Available bandwidth varies from 0 to maximum bandwidth of wireless channel (Figure 4). When available bandwidth is less than Low-threshold, we cannot use the advantage of FG-LIW TCP but still have the same performance as traditional TCP's (interval T). But if

```
if (High_threshold ≤ Available_BW)
    Init_window_size = 4;    /* high utilization */
else if (Middle_threshold ≤ Available_BW < High_threshold)
    Init_window_size = 3;
else if (Low_threshold ≤ Available_BW < Middle_threshold)
    Init_window_size = 2;
Else
    Init_window_size = 1;    /* traditional TCP operation */
```

Fig. 3. FG-LIW Decision Function

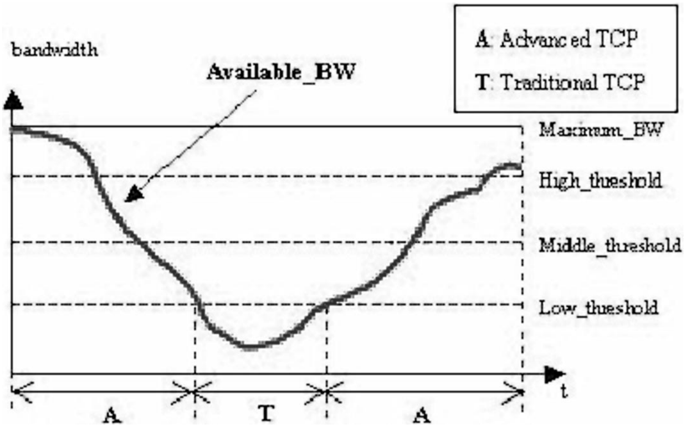


Fig. 4. Available-BW and three-level-threshold

available bandwidth is greater than Low-threshold, simulation gives us improved performance result (interval A).

5 Simulation

The topology of the simulated network is given in figure 5. Even though this topology does not mimic all the characteristics of any specific wireless environment network, it gives a clear picture of the improving performance of FG-LIW. As we can see from the figure 5, transmission throughput over wireless network will be dominated by the capacity of shared wireless link(in here, 1M shared link). In the topology, network is consisted of 5 TCP source nodes, 5 intermediate nodes, 5 destination nodes and base station. Source nodes, s1 and s2, are supposed to have FG-LIW function and to employ delayed ACK policy while s3 to s5 to have Non-FG-LIW function and to acknowledge every segment. All source nodes are based on TCP Reno and have 200ms retransmission timeout, 16 for slow start threshold. Wired links have characteristics of 1ms propagation delay, 50M[bps] bandwidth while wireless link has 3ms propagation delay, 1M[bps] shared bandwidth. We can see FG-LIW implemented node(s1, s2)'s throughput

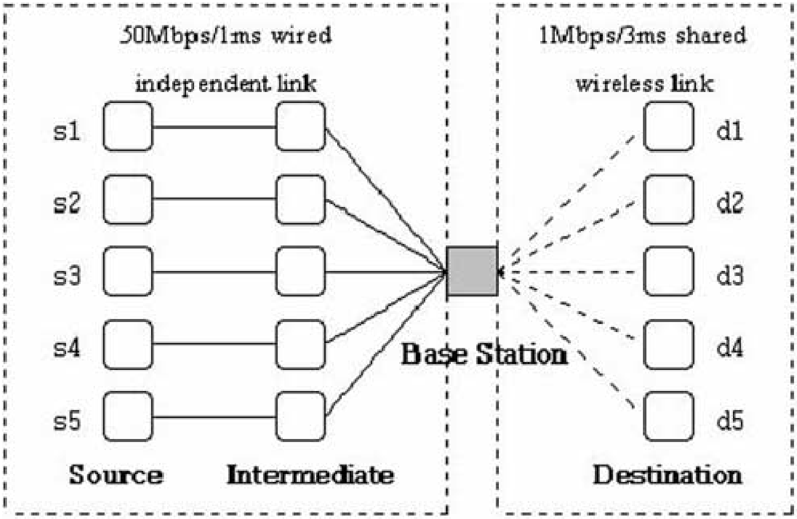


Fig. 5. Simulated Network Topology

Table 1. Simulation Result when 2 FG-LIW TCP flows and 3 non FG-LIW TCP flows share 1M bandwidth

Initial cwnd size	A FG-LIW TCP flow		B FG-LIW TCP flow		Non FG-LIW TCP traffic		
	20k transfer time[sec]	Throughput [pkt/sec]	20k transfer time[sec]	Throughput [pkt/sec]	C TCP threshold [pkt/sec]	D TCP threshold [pkt/sec]	E TCP threshold [pkt/sec]
1	0.4682	42.72	0.4960	40.32	225.28	224.35	221.90
2	0.0958	208.68	0.1454	137.51	232.38	231.91	227.00
3	0.0802	249.32	0.1201	166.49	239.33	229.72	229.72
4	0.0642	311.74	0.1081	184.93	239.05	223.44	231.50

improvement without degradation of existing Non-FG-LIW traffics(s3, s4 and s5) when LIW varies from 1 to 4. Simulated transfer data size of FG-LIW is 20K[bit] and BS uses round-robin scheduling scheme for the shared 1M wireless link. Each maximum segment size of 5 independent TCP connections is used as 1000[bit] for simplicity of calculation.

Our simulation result shows that our modification can lead to better performance, especially for short flows. We can see decreased transfer time and improved throughput with increase from 1 to 4, in the FG-LIW(Table1) guaranteeing existing normal other traffics(C,D and E). FG-LIW does not affect other traffics because FG-LIW utilizes only unused available bandwidth.

6 Conclusion

The idea of FG-LIW is to maximize the bandwidth utilization as long as there is usable bandwidth. Our suggestion provides a significant improvement of TCP throughput without degradation of other traffics. This scheme can be ported while the infrastructure of network needs minimal changes. Even in the worst case, FG-LIW still maintains the same performance as the traditional TCP mechanism. Our proposal controls TCP behavior dynamically according to network condition. Especially, for short lived flows, it is very useful scheme.

References

1. N. Cardwell, S. Savage, and T. Anderson: Modeling the Performance of Short TCP Connections. Technical Report. Computer Science Department, Washington University. (Nov. 1998)
2. H. Balakrishnan et al: Improving TCP/IP Performance Over Wireless Networks. Proc. 14th Intl. Conf. Distributed Computing Systems. (Jun. 1994) 12–20
3. M. Allman: On the Generation and Use of TCP Acknowledgements. ACM Computer Communications Review. **28** (Oct. 1998)
4. M. Allman, V. Paxson, and W. R. Stevens: TCP Congestion Control. RFC2581. (Apr. 1999)
5. M. Allman, S. floyd: Increasing TCPs Initial Window. RFC2414. (Sept. 1998)
6. R. Braden: Requirements for Internet Hosts Communication Layers. RFC1122. (Oct. 1989)
7. Douglas E. Comer: Internetworking with TCP/IP. Fourth Edition. Prentice Hall
8. Kostas Penticousis: TCP In Wired-Cum-Wireless Environments. IEEE Communications Surveys. Fourth Quarter 2000

A Comparative Study on the Performance of Detectors and Interference Models for OCDMA LANs*

Yongchul Yun[1], Jinwoo Choe[1], and Wonjin Sung[1]

Department of Electronic Engineering, Sogang University,
1 Sinsu-dong, Mapo-gu, Seoul 121-742, Korea
{notavoid,xinu,wsung}@sogang.ac.kr

Abstract. 2-D OCDMA is considered to be a viable technical solution for optical LANs, and a considerable amount of research effort has been devoted to various 2-D OCDMA techniques. In this paper, we propose two new interference models for 2-D OCDMA LANs employing unipolar random codes, and derive maximum-likelihood detectors based on these interference models. The BER performance of the maximum likelihood detectors and that of other existing detectors are compared through extensive computer simulations. In addition, the complexity of high-speed implementation of the detectors is assessed, and as a result, we found that the AND detector and the maximum-likelihood detectors for the pulse-binomial and the pulse-Poisson model offer the best trade-off between the BER performance and the facility of high-speed implementation.

1 Introduction

Advances in the optical communication technology enabled it to take over the role of the electronic communication technology in long-haul communication systems. On the other hand, it is relatively recent that visible research efforts to extend the benefit of the optical communication technology to the scale of *Local Area Network* (LAN) were initiated, and the electronic communication technology still remains to be the dominant technology for short-haul communication. The delayed introduction of the optical communication technology in the realm of local area networking, might be a result of the relatively low bandwidth requirement of LANs. However, with the advent and expansion of "bandwidth-greedy" network applications such as multimedia applications and 3-D graphics applications, LANs built upon the electronic communication technology (with shared bandwidth typically ranging from 10 Mbps to 1 Gbps) may experience serious congestion, and become a bottleneck eventually. Therefore, the demand for the optical LAN technology is expected to increase steeply in the near future, and the time of large scale deployment of optical LANs will heavily depend on their affordability and operational facility.

* This work was supported by the Korea Science Engineering Foundation under Grant No. R01-2001-00542.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 223–238, 2002.

While *Time Division Multiplexing* (TDM) and *Wavelength Division Multiplexing* (WDM) have been driving the astonishing success of the optical communication technology, they may not provide practical solutions for LANs preferable to the existing copper-wire-based solutions [1]. This is because switching packets at transmission rates over several Gbps is still challenging, and even if it may be attainable via the state-of-the-art optical and/or electronic communication technology, the cost is likely to be far beyond typical budget for LANs. Further, fully optical switching techniques such as wavelength-based switching techniques and burst switching techniques, require global-scale dimensioning and sophisticated resource allocation schemes, and hence, may not provide a viable solution for LANs where the network configuration may change frequently and the traffic patterns are highly irregular. For these reasons, multi-access techniques that have been originally developed for wireless and other types of shared medium communication, are revaluated in the field of optical communications [2,3,4,5,6,7], and in particular, *Optical Code Division Multiple Access* (OCDMA) is attracting considerable amount of research interest [3,4,5,6].

Due to technical difficulties in implementing optical phase modulation /detection, intensity (or amplitude) modulation/detection is widely employed in OCDMA. A general class of OCDMA where spreading codes are constructed by placing optical pulses at different positions along wavelength and time axes, is often referred to as *two dimensional OCDMA* (2-D OCDMA) or *multi-wavelength* OCDMA (MW-OCDMA) [8,9,10,11,12]. Fundamental technical problems in 2-D OCDMA include code design problems [9,10,11], detection problems [8], and implementation issues [13,14,15], among which the detection problem is the main focus of this paper. In the literature, a number of results and proposals on the OCDMA detection problem can be found [8,16], and on account of high-speed transmission in optic fiber networks, relatively simple detectors are mainly considered. However, in [8] it was shown that these simple detectors may degrade the bandwidth efficiency significantly under OCDMA systems adopting 2-D pulse-position modulation (i.e., On-Off keying with unipolar 2-D OCDMA codes), and an ML detector based on an stochastic interference model was proposed for improved bandwidth efficiency.[1] In this paper, we extend the study on the 2-D OCDMA detection problem in [8] by introducing two additional interference models for 2-D OCDMA that are different from the interference model used in [8]. The *Maximum-Likelihood* (ML) detectors for these new model will be derived, and their BER performance will be investigated. In addition, several suboptimal detectors with relatively simple structure will also be considered and compared to the ML detectors. The objective of our study is to evaluate a wide range of 2-D OCDMA detectors in terms of their BER performance and anticipated complexity of implementation.

[1] The ML detector proposed in [8] degenerates to logical AND operation when the pulse density is low, for which the authors refer to the ML detector as “AND” detector. The BER performance analysis for the ML detector in [8] is actually carried out for this degenerated ML detector (i.e., for the “real” logical AND detector introduced in [16]).

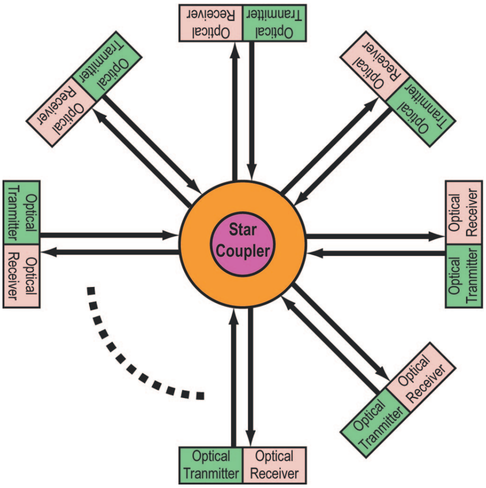


Fig. 1. Logical topology of typical OCDMA LANs

The rest of the paper is organized as follows. In Section 2, the system model and three different interference models will be described. In Section 3, the ML detectors for different interference models will be obtained, and a set of detectors with relatively simple structure will also be introduced. In Section 4, the BER performance of the detectors will be compared through extensive simulations under various 2-D OCDMA configurations. Finally, in Section 5, the complexity and practical value of the detectors will be assessed with implementation issues taken into account, and the paper is brought to a conclusion.

2 System and Interference Model

2.1 System Model

In this paper, we suppose the same OCDMA LAN topology that was studied in [1,8]; i.e., all network nodes are assumed to be interconnected through one or more passive star coupler as illustrated in Fig. 1. Consequently, every receivers will receive optical signals injected into network by all transmitters. Further, incoherent light sources and ideal couplers (i.e., 0 dB attenuation and no additive or multiplicative channel noise) are assumed in the system model, so that the received signal can be expressed as the superposition of all transmitted optical signals. This implies that the *multi-access interference* (MAI) is the only source of detection errors.

General 2-D OCDMA codes can be represented by $L \times M$ matrices illustrated in Fig. 2, where L and M represent the number of wavelengths used for OCDMA and the spreading factor along the time axis (i.e., the ratio of the bit period to the chip period), respectively. We assume On-off keying is used with 2-D OCDMA codes for the transmission of each bits; that is, when “0” is sent, no optical

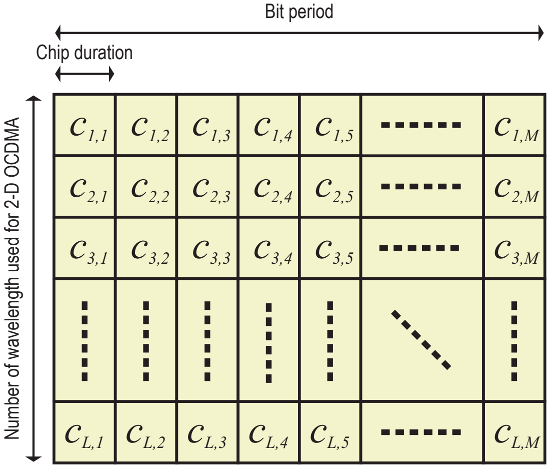

Bit period
Chip duration
Number of wavelength used for 2-D OCDMA
$c_{1,1}$ $c_{1,2}$ $c_{1,3}$ $c_{1,4}$ $c_{1,5}$ $c_{1,M}$
$c_{2,1}$ $c_{2,2}$ $c_{2,3}$ $c_{2,4}$ $c_{2,5}$ $c_{2,M}$
$c_{3,1}$ $c_{3,2}$ $c_{3,3}$ $c_{3,4}$ $c_{3,5}$ $c_{3,M}$
$c_{L,1}$ $c_{L,2}$ $c_{L,3}$ $c_{L,4}$ $c_{L,5}$ $c_{L,M}$

Fig. 2. Matrix representation of 2-D OCDMA codes and signals

pulse is transmitted, and when “1” is sent, a pulse with intensity $c_{i,j}$ will be transmitted during the j-th chip period over the i-th wavelength. In addition, the following conditions will be assumed throughout the paper, which are also a part of basic assumptions in [8].

- 2-D OCDMA codes are randomly generated at the transmitter in such a way that all codes consist of an identical number pulses of unit intensity, and thus, can be identified only by their pulse position, and ; i.e., $c_{i,j} = 0$ or 1 and $\sum_{(i,j)\in E} c_{i,j} = W$ for all codes, where $E = \{1, 2, \ldots, L\} \times \{1, 2, \ldots, M\}$, and W is a constant called the weight of the 2-D OCDMA code set.
- All communicating transmitter-receiver pairs are under perfect bit synchronization.
- All signals received by a receiver (i.e., including interfering signals) are chip-synchronized.

2.2 Interference Model

In this section, we introduce three different models for the interfering signals: *User-Binomial* (UB) model, *Pulse-Binomial* (PB) model, and *Pulse-Poisson* (PP) model. Although a number of idealistic assumptions were made in the system model, deriving the exact probabilistic model for the interfering signal is still a challenging task. Therefore, further simplifications and/or approximations of internal activities are required, and these models are obtained through different simplification and/or approximation steps.

User-Binomial Model. The UB model is the interference model used in [8]. In this model, each interfering transmitter will transmit a pulse of unit intensity at a position $(i, j) \in E$ with probability $\frac{W}{2LM}$. The factor $\frac{1}{2}$ comes from the fact that "1" will be sent with probability $\frac{1}{2}$ and $\frac{W}{LM}$ from the fact that a transmitter will send W pulses at different positions in the $L \times M$ code matrix. Consequently, assuming the independence among interfering transmitters, we can derive the probability distribution of $X_{i,j}$, the intensity of interfering pulse at position (i, j) as the followings binomial distribution:

$$\mathbb{P}[X_{i,j} = x_{i,j}] = \binom{N_I}{x_{i,j}} \left(\frac{W}{2LM}\right)^{x_{i,j}} \left(1 - \frac{W}{2LM}\right)^{N_I - x_{i,j}}, \tag{1}$$

where N_I is the number of interfering transmitters. In the actual 2-D OCDMA system described in the previous section, the intensities of interfering pulses at different positions in the matrix $X = [X_{i,j}]$ are stochastically dependent, because each transmitter is allowed to send at most W pulses at different positions. However, when N_I or W is sufficiently large, the dependence is expected to be fairly imperceptible. Therefore, if we assume the independence of $\{X_{i,j} | (i, j) \in E\}$, the joint distribution of the interference matrix X immediately follows as

$$\mathbb{P}[X_{i,j} = x_{i,j}; E] = \prod_{(i,j) \in E} \binom{N_I}{x_{i,j}} \left(\frac{W}{2LM}\right)^{x_{i,j}} \left(1 - \frac{W}{2LM}\right)^{N_I - x_{i,j}}, \tag{2}$$

where $x = [x_{i,j}]_{(i,j) \in E}$.[2] Now, let C be the 2-D OCDMA code that a receiver is supposed to receive when a bit "1" is sent to the receiver by a transmitter and E_1 be the subset of E where the receiver is supposed to detect a pulse when C is transmitted by the transmitter; i.e., $E_1 = \{(i, j) \in E | c_{i,j} = 1\}$. Then, again from the independence assumption, the joint distribution of pulse intensities at E_1 can be derived in the same way as

$$\mathbb{P}\left[X_{i,j} = x_{i,j}; E_1\right] = \prod_{(i,j) \in E_1} \binom{N_I}{x_{i,j}} \left(\frac{W}{2LM}\right)^{x_{i,j}} \left(1 - \frac{W}{2LM}\right)^{N_I - x_{i,j}}. \tag{3}$$

Pulse-Binomial Model. In the second model that is named PB model, the fundamental assumption is that each pulse in interfering signals will be independently turned on and off with probability $\frac{1}{2}$, and its position will also be independently chosen from E. Since there are $N_I W$ pulses that may be turned on and off independently and LM different positions where a pulse may be located, using combinatorial methods we get

$$\mathbb{P}[X_{i,j} = x_{i,j}; E] = \frac{s!}{(LM)^s \prod_{(i,j) \in E} x_{i,j}!} \frac{\binom{N_I W}{s}}{2^{N_I W}}, \tag{4}$$

where $s := \sum_{(i,j) \in E} x_{i,j}$. Similarly, the joint distribution of interfering pulse intensities at positions E_1 can be derived as

[2] Note that although the independence assumption may be a reasonable simplification, the total pulse intensity of interfering signals may become larger than $N_I W$ in the UB interference model.

$$\mathbb{P}\left[X_{i,j} = x_{i,j}; E_1\right] = \frac{s_1!}{W^{s_1} \prod_{(i,j)\in E_1} x_{i,j}!} \binom{N_I W}{s_1} q^{s_1} \left(1-q\right)^{N_I W - s_1} , \quad (5)$$

where $s_1 := \sum_{(i,j)\in E_1} x_{i,j}$ and $q := \frac{W}{2LM}$.[3] Note that the advantage of the PB model over the UB model is that the interdependence among interfering pulse intensities at different positions can reflected; i.e., $\{X_{i,j} : (i,j) \in E\}$ are not assumed to be independent anymore and the total intensity of interfering pulses is limited by $N_I W$. However, there is a cost for this advantage, that is, the total intensity of interfering pulses may not be a multiple of W and the pulse intensity at a single position may exceed N_I.

Pulse-Poisson Model. In the PB model, the interference signal is generated as if each of $N_I W$ independent potential sources of interfering pulses adds a single pulse of unit intensity at a random position in E. One way to make $\{X_{i,j} : (i,j) \in E\}$ mutually independent in the PB model is to increase the population of such potential sources of interference to infinity while reducing the probability of each source being active (i.e., the probability that a source actually generates a pulse) to 0. In other words, let N and p_a be the population of potential sources of (single pulse) interference and the probability that a source generates a pulse at a position in E, respectively. The joint distribution of interfering pulse intensities can then be derived as

$$\mathbb{P}[X_{i,j} = x_{i,j}; E] = \frac{s!}{(LM)^s \prod_{(i,j)\in E} x_{i,j}!} \binom{N}{s} p_a^s (1-p_a)^{N-s}.$$

Now, if N is sent to infinity while keeping the product $p_a N$ fixed at $\alpha > 0$, the joint distribution will converge to

$$\mathbb{P}[X_{i,j} = x_{i,j}; E] = e^{-\alpha} \frac{\left(\frac{\alpha}{LM}\right)^s}{\prod_{(i,j)\in E} x_{i,j}!}. \quad (6)$$

This indicates that when there are a large number of nodes in the star topology, each of which transmits either "0" or "1" with relatively small probability, the interference signal can be accurately characterized by independent Poisson random variables representing the pulse intensities at different positions in E. In fact, it is a fairly usual situation in LANs that only a small portion of hundreds of nodes are transmitting information.

The parameter α can be interpreted as the pulse density in the interference signal measured in the unit of pulse/bit-period. Also, note that from the independence of pulse intensities at different positions, the joint distribution of interfering pulse intensities at positions E_1 can be derived as

$$\mathbb{P}[X_{i,j} = x_{i,j}; E_1] = e^{-\frac{\alpha W}{LM}} \frac{\left(\frac{\alpha}{LM}\right)^{s_1}}{\prod_{(i,j)\in E_1} x_{i,j}!}. \quad (7)$$

[3] This model was implicitly assumed in [8] for the estimation of the BER performance (i.e., Eq. (10)) of a simple detector.

3 OCDMA Detectors

3.1 Maximum Likelihood Detectors

Let R be the pulse intensity matrix received by a optical receiver, and C be the 2-D code that the receiver is supposed to detect. Then, depending on whether "0" or "1" is sent, the received intensity matrix R may be composed of interference signals superimposed on the code C or only interference signals; i.e.,

$$R = \begin{cases} X & \text{if "0" is sent,} \\ C + X & \text{if "1" is sent.} \end{cases}$$

From this relationship between R and X, the conditional distribution of R can be expressed in terms of the joint distribution of the interfering pulse intensities as follows.

$$\begin{aligned} \mathbb{P}[R = r | \text{"0" is sent}] &= \mathbb{P}[X = r], \text{ and} \\ \mathbb{P}[R = r | \text{"1" is sent}] &= \mathbb{P}[X = r - C]. \end{aligned}$$

Therefore, the ML decision rule can be stated as

$$\begin{aligned} &\text{"0" is received if } \mathbb{P}[X = r] > \mathbb{P}[X = r - C], \\ &\text{"1" is received if } \mathbb{P}[X = r] < \mathbb{P}[X = r - C]. \end{aligned} \tag{8}$$

Since the decision rule contains the joint distribution of interfering pulse intensities, the actual form of the ML detector will depend on the interference model used, and in the following, the ML decision rules for three different models for interference are derived.

ML detector for the UB interference model

$$\prod_{(i,j)\in E_1} \frac{r_{i,j}}{N_I - r_{i,j} + 1} \underset{\text{"0 detected"}}{\overset{\text{"1 detected"}}{\gtrless}} \left(\frac{W}{2LM - W}\right)^W . \tag{9}$$

ML detector for the PB interference model

$$\frac{(N_I W - s)!}{((N_I + 1)W - s)!} \prod_{(i,j)\in E_1} r_{i,j} \underset{\text{"0 detected"}}{\overset{\text{"1 detected"}}{\gtrless}} (LM)^{-W}, \tag{10}$$

where $s = \sum_{(i,j)\in E} r_{i,j}$.

ML detector for the PP interference model

$$\prod_{(i,j)\in E_1} r_{i,j} \underset{\text{"0 detected"}}{\overset{\text{"1 detected"}}{\gtrless}} \left(\frac{\alpha}{LM}\right)^W . \tag{11}$$

At a glance, the ML detectors for UB, PB, and PP interference models may appear different in their expressions. However, when N_I is large and the code weight W is set to a small value compared to LM, they are not fundamentally different in the sense that their decision rules are essentially based on the value of $\prod_{(i,j)\in E_1} r_{i,j}$, the product of pulse intensity at positions E_1. To see this, first note that typical values of $r_{i,j}$ are usually much smaller than N_I. Therefore, we may approximate $N_I - r_{i,j} + 1 \approx N_I$ and $W/(2LM - W) \approx W/2LM$ in (9), and the decision rule for the UB model can then be simplified to

$$\prod_{(i,j)\in E_1} r_{i,j} \underset{\text{“0 detected”}}{\overset{\text{“1 detected”}}{\gtrless}} \left(\frac{N_I W}{2LM}\right)^W . \tag{12}$$

Here, it should be recognized that $N_I W/2$ may be interpreted as the interference pulse density in the UB model. Therefore, this approximated ML detector is essentially equivalent to the ML detector for the PP model.

Similar observation can be made in the ML detector for the PB model. Note that

$$\frac{(N_I W - s)!}{((N_I + 1)W - s)!} = \frac{(N_I W)^{-W}}{\left(1 - \frac{s}{N_I W} + \frac{1}{N_I W}\right)\cdots\left(1 - \frac{s}{N_I W} + \frac{1}{N_I}\right)} .$$

Further, from the law of large numbers we can see that $\frac{s}{N_I W}$ should be close to $1/2$, and hence, we may approximate

$$\left(1 - \frac{s}{N_I W} + \frac{1}{N_I W}\right)\cdots\left(1 - \frac{s}{N_I W} + \frac{1}{N_I}\right) \approx 2^{-W} .$$

This indicates that the ML detector for the PB model can also be approximated by (12).

3.2 Simple Detectors

The LHS and RHS of (9)–(11) can be thought of as the statistics of the received signal R and the threshold to which the statistics will be compared to, respectively. Inspecting the statistics of R involved in the ML decision process, one can easily verify that at least $W - 1$ multiplications are required to compute these statistics from R. Therefore, although the ML detectors given in (9)–(11) are expected to yield optimal or close-to-optimal BER performance, applying them to a data stream of several Gbps may not be a simple task. For this reason, a couple of simple detectors, that is, AND and SUM detectors have been proposed, and their performance was compared to that of ML detectors [8]. The original AND detector is first introduced in [16], and as its name stands for, the decision process of the AND detector is identical to the logical AND operation. In other words, if pulses of any intensity levels are detected at every position in E_1, the AND detector supposes that “1” is received, and otherwise, “0” is received.

AND detector

$$\sum_{(i,j)\in E_1} \lceil r_{i,j} \rceil \begin{array}{c}\scriptstyle\text{“1 detected”}\\ = \\ \scriptstyle\text{“0 detected”} \\ /= \end{array} W, \tag{13}$$

where $\lceil x \rceil = \max\{x, 1\}$.

On the contrary, the SUM detector uses arithmetic “SUM” operation on the pulse intensities received at the positions E_1. In other words, the total intensity of pulses at positions E_1 is greater than or equal to W then the conventional SUM detector assumes that “1” is received, and otherwise, it assumes “0” is received.

SUM detector

$$\sum_{(i,j)\in E_1} r_{i,j} \begin{array}{c}\scriptstyle\text{“1 detected”}\\ \geq \\ \scriptstyle\text{“0 detected”} \\ < \end{array} W. \tag{14}$$

In [8], it has been shown that the AND detector exhibits better BER performance than the SUM detector. However, it should be noted that W is not the optimal threshold for the SUM detector, and by optimizing the threshold, the BER performance of the SUM detector may be improved. To find the optimal threshold, we first need to derive the conditional distribution of $s_1 = \sum_{(i,j)\in E_1} r_{i,j}$. If the PP model is assumed for the interference, one can easily verify that

$$\mathbb{P}\left[\sum_{(i,j)\in E_1} r_{i,j} = s_1 \middle| \text{“0” is sent} \right] = e^{-\frac{\alpha W}{LM}} \frac{\left(\frac{\alpha}{LM}\right)^{s_1}}{s_1!} \quad \text{for } s_1 = 0, 1, \ldots, \text{ and}$$

$$\mathbb{P}\left[\sum_{(i,j)\in E_1} r_{i,j} = s_1 \middle| \text{“1” is sent} \right] = e^{-\frac{\alpha W}{LM}} \frac{\left(\frac{\alpha}{LM}\right)^{s_1 - W}}{(s_1 - W)!} \quad \text{for } s_1 = W, W+1, \ldots.$$

Therefore, the ML decision rule based on the statistic $\sum_{(i,j)\in E_1} r_{i,j}$ will be given as follows.

Optimized SUM detector

$$\sum_{(i,j)\in E_1} r_{i,j} \begin{array}{c}\scriptstyle\text{“1 detected”}\\ \geq \\ \scriptstyle\text{“0 detected”} \\ < \end{array} T_o, \tag{15}$$

where T_o denotes the largest real root of the equation

$$x(x-1)(x-2)\cdots(x-W+1) = \left(\frac{\alpha}{LM}\right)^W. \tag{16}$$

Even with the optimized thresholds, the SUM detector possesses a critical weakness. Note that if no pulse is detected at any of the positions E_1, there is obviously no chance that “1” is transmitted. This fact is reflected in the ML detectors (9)–(11) and the AND detector, and hence, they never makes erroneous decision when $r_{i,j} = 0$ for at least one position $(i,j) \in E_1$. On the other hand, both the conventional SUM detector and the optimized SUM detector may make such erroneous decision with a positive probability. A simple way to prevent SUM detectors from making such obvious misjudgment is to combine them with the AND detector. In other words, we may use the AND detector to filter out the cases where there is no chance that "1" is transmitted, and then use the SUM detector as a mean to confirm the decision made by the AND detector. For instance, combining the AND detector and the optimized SUM detector will result in the following decision rule.

Optimized AND-SUM detector

$$\begin{array}{lll} \text{“1 detected”} & \text{if} & \displaystyle\sum_{(i,j)\in E_1} \lceil r_{i,j} \rceil = W \quad \text{and} \quad \sum_{(i,j)\in E_1} r_{i,j} \geq T_o, \\ \text{“0 detected”} & \text{otherwise.} & \end{array} \tag{17}$$

4 Numerical Experiments

In this section, we compare the BER performance of various 2-D OCDMA detectors discussed in Section 3 through an extensive set of numerical experiments. For the sake of tidy and orderly presentation of the simulation results, the abbreviations in Table 1 for the names of 7 different detectors will be used throughout this section including both figures and tables.

Table 1. The abbreviations of 7 detector names used in Section 4

AND	the AND detector given by (13)
SUM	the SUM detector given by (14)
OPTSUM	the SUM detector with the optimized threshold given by (15)
ANDSUM	the combined AND-SUM detector given by (17)
MLUB	the ML detector for the UB interference model given by (9)
MLPB	the ML detector for the PB interference model given by (10)
MLPP	the ML detector for the PP interference model given by (11)

In the experiments, $L = 10$ wavelengths are used for 2-D OCDMA, and the number of chip periods in a bit period (i.e., M) has been set to two different values, 10 and 20. Therefore, the corresponding number of pulse positions in E is computed to be 100 and 200, respectively. For both cases, we changed W, the weight of codes to 6 different values, (i.e., $W = 2, 4, 7, 10, 15, 20$), and for each value of W, we measured the BER through computer simulations for different number of active nodes.[4] For each sample point in the simulation results, the duration of simulation has been determined in such a way that the 99%

[4] Here, "active nodes" means the nodes that are currently transmitting bit streams over the shared optical medium. Consequently, if there are N_a active nodes, then from the view point of individual receiver, the effect is as if there are $N_I = N_\mathrm{a} - 1$ independent sources of interferences in the LAN.

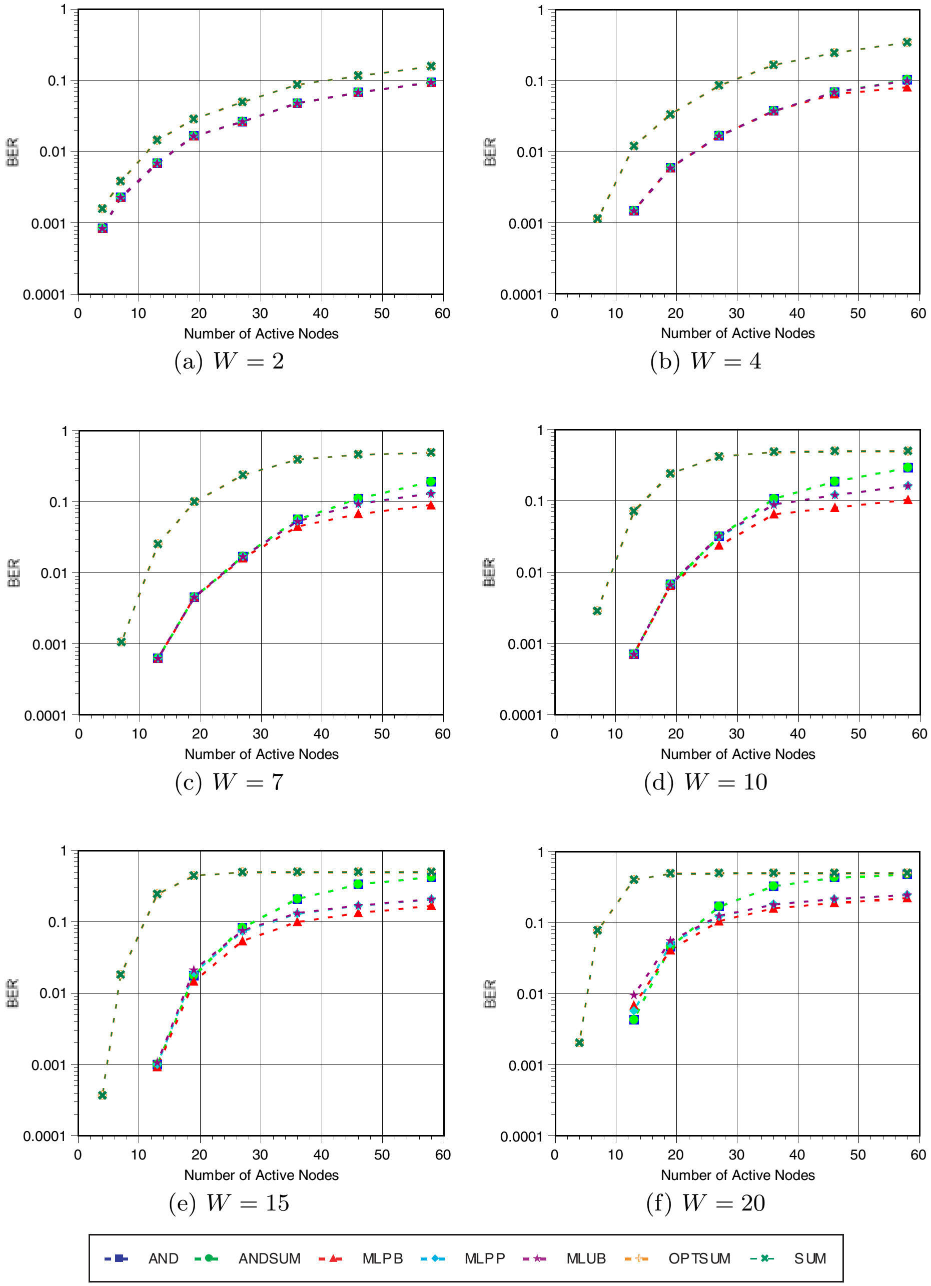
confidence interval of the BER does not exceed 20% of the estimated value of BER. For the ML detectors, it is assumed that the number of interfering nodes (or the number of active nodes) is known beforehand, and also, the value of α in the MLPP detector is set to the expected density of pulses during a bit period given N_I; i.e.,

$$\alpha = \frac{N_I W}{2}.$$

Fig. 3 and Fig. 4 show the simulation results for $L = 10$, $M = 10$ and for $L = 10$, $M = 20$, respectively. From the figures, one can observe that 7 detectors can be grouped into 4 groups exhibiting similar BER performance. It turned out that the SUM detector and the OPTSUM detector show nearly identical BER performance, which is the worst among the 4 groups. It may appear somewhat surprising that optimizing the threshold in the SUM detector does not yield any gain in the BER performance. However, this is because the optimal threshold T_o, which can be obtained by solving (16), takes value between $W-1$ and W unless the interfering pulse density $N_I W/2$ is significantly larger than LM, the number of pulse positions in E. Consequently, we found that the OPTSUM detector degenerates to the SUM detector when their BER is relatively small. When the pulse density is larger enough to increase T_o beyond W, marginal improvements in the BER performance over the original SUM detector can actually be observed in the simulation results. However, the BER in such heavy interference situations is close to 0.5, and the marginal improvements in the BER performance is neither practically meaningful nor clearly visible.

The second group of detectors showing the next worst BER performance, includes the AND detector and the ANDSUM detector. Note that $\sum_{(i,j)\in E_1} \lceil r_{i,j} \rceil < W$ implies that $\sum_{(i,j)\in E_1} r_{i,j} < W$. This indicates that the SUM detector part in the ANDSUM detector plays any role only if the optimal threshold T_o is larger than the code weight W. Therefore, the ANDSUM detector will degenerate to the AND detector at low interfering pulse density for the reason discussed in the previous paragraph, and hence, the AND detector and the ANDSUM detector exhibit nearly identical BER performance. The BER performance of the AND and ANDSUM detectors is comparable to that of the ML detectors at low interfering pulse density, and gradually becomes inferior as the number of active nodes increases.

The ML detectors exhibit lower BER than other simple detectors over the whole range of W and N_I, and demonstrated BER performance comparable to each other as expected in Section 3.1. In particular, the BER of the MLUB detector and that of the MLPP detector are indistinguishable as can be observed in Fig. 3 and Fig. 4, and hence, the MLUB and MLPP detectors constitute the third group of curves in the figure. The MLPB detector, on the other hand, provides slightly (but visibly) improved BER performance over the MLPP and MLUB detectors. Remember that the PB interference model account for the interdependency between pulse intensities $r_{i,j}$ at different pulse positions (i,j) in E, while the UB mode and the PP model fundamentally assume the independence of $\{r_{i,j} | (i,j) \in E\}$. As a result, only the pulse intensities at $(i,j) \in E_1$



(a) $W = 2$

(b) $W = 4$

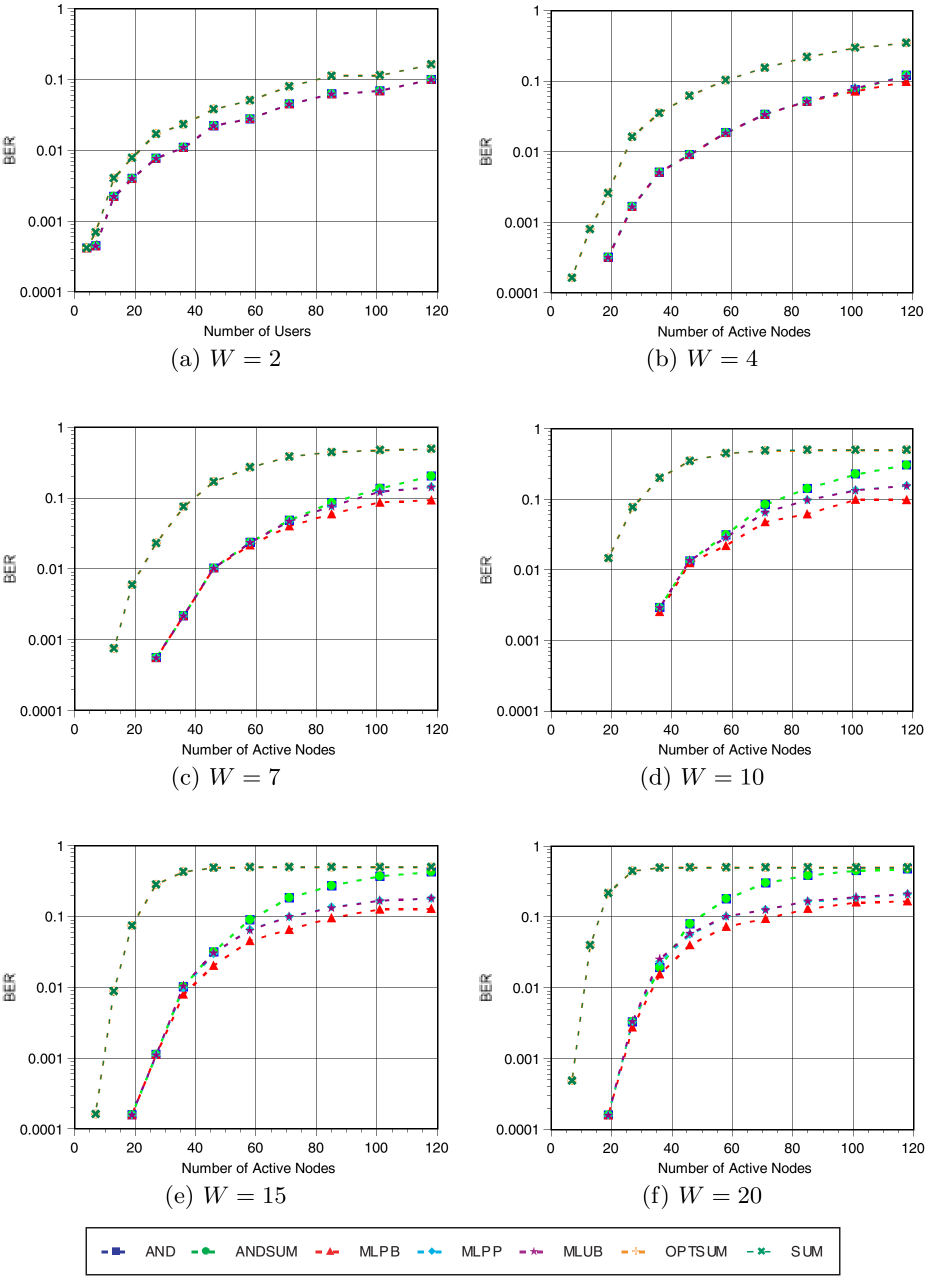
(c) $W = 7$

(d) $W = 10$

(e) $W = 15$

(f) $W = 20$

Fig. 3. Comparison of the BER performance of 7 detectors for $L = 10$ and $M = 10$



(a) $W = 2$

(b) $W = 4$

(c) $W = 7$

(d) $W = 10$

(e) $W = 15$

(f) $W = 20$

Fig. 4. Comparison of the BER performance of 7 detectors for $L = 10$ and $M = 20$

are involved in the decision process of the MLUB and MLPP detectors, while the MLPB detector makes decision on the pulse intensities at all positions in E. In a practical 2-D OCDMA system, there actually exists interdependency of a certain degree between pulse intensities at different positions. Therefore, the improved BER performance of the MLPB detector is believed to be achieved by taking the interdependency into account in the decision process.

In summary, simulation results show that 7 detectors can be ordered according to their BER performance as follows:

Low pulse density (i.e., $\frac{N_I W}{2LM} < 1$)

$$\text{MLPB} \approx \text{MLPP} \approx \text{MLUB} \approx \text{AND} \approx \text{ANDSUM} < \text{OPTSUM} \approx \text{SUM}.$$

High pulse density (i.e., $\frac{N_I W}{2LM} > 1$)

$$\text{MLPB} < \text{MLPP} \approx \text{MLUB} < \text{AND} \approx \text{ANDSUM} < \text{OPTSUM} \approx \text{SUM}.$$

5 Conclusion

In this paper, we introduced two new interference models, the PB and the PP models for 2-D OCDMA systems, which can replace the UB model used in [8] for the stochastic characterization of interfering signals in 2-D OCDMA LANs. The ML detectors for these new interference models (i.e., the MLPB and MLPP detectors) are derived, and their BER performance is compared to that of the MLUB detector through extensive numerical experiments. In addition, a set of detectors with relatively simple structure including the traditional AND and SUM detectors were introduced, and their BER performance was investigated and compared to that of ML detectors through simulations. As a result, it was found that the MLPB detector consistently exhibits BER performance superior to the other detectors, and the MLUB and MLPP detectors closely follow the MLPB detector. Among the detectors with relatively simple structure, the AND detector and its variation (i.e., the combined AND-SUM detector) turned out to have BER performance comparable to the ML detectors. In particular, the BER performance of the AND detector matches that of the ML detectors at low pulse density (i.e. when $\frac{N_I W}{2LM} < 1$).

Given the BER performance of the 7 detectors considered in this paper, assessing their cost and complexity for implementation is the next step of course. In general, the 2-D OCDMA detectors can be implemented in two main parts: *statistics extractor* and *thresholder*. At the statistics extractor stage, a statistical figure will be computed from the received signal, and will be passed to the thresholder. The thresholder will then decide whether “0” or “1” is received by comparing the statistics to a threshold which may vary over relative long time periods, e.g., as the number of active nodes changes. Although such a detection process will take place once for each bit period, for the synchronization with the transmitter, the statistics extractor and the thresholder are usually required to operate at the chip rate, which may be higher than the bit rate by orders of magnitude. Comparing a time-sampled signal to a fixed (or slowly varying) threshold is a fairly simple operation, and hence, can be carried out at a fairly high speed. This means that the statistics extractor is likely to become a bottleneck in the signal path and determine the maximum detection rate of a OCDMA detector. Consequently, the SUM detector and the AND detector (including their variations) are deemed to have advantage over the ML detectors because the statistics of the received signal (i.e., the left hand side of (13), (14), (15), and (17)) can be obtained from the received signal using relatively simple hardwired optical and/or electronic devices composed of mixer and hard-limiter [17,16]. On the other hand, extracting the statistics required by the ML detectors at the chip rate is not a simple task, and may not be feasible even with the state-of-the-art electrical and/or optical signal processing technology. This is partially because the statistics contains the product of the pulse intensity. However, in the case of the MLUB and MLPB detectors, a more critical problem is that the statistics contains N_I, the number of interfering nodes, which typically fluctuates considerably with time. In other words, the MLUB and MLPB detectors will require different statistics of the received signal moment by moment, and the statistics extractor must be designed to provide such flexibility. This makes it more difficult to maximize the processing speed of the MLUB and MLPB detectors using hardwired devices. As a result, the MLPP detector should be considered more suitable for high speed implementation than the MLUB and MLPB detectors.

Summarizing the studies on the 7 detectors carried out in this paper, we can conclude that the AND, MLPB, and MLPP detectors form a lineup of OCDMA detectors that offer the best trade-off between the BER performance and the facility of high-speed implementation as described in the following.

AND detector

- Shows BER performance close to the ML detectors under low interference ($\frac{N_I W}{2LM} < 1$) and moderate BER performance under high interference ($\frac{N_I W}{2LM} > 1$).
- Appropriate for high-speed implementation.

MLPP detector

- Shows the best BER performance under low interference and the second best BER performance under high interference ($\frac{N_I W}{2LM} > 1$).
- Requires devices that can effectively multiply received pulse intensities for high-speed implementation.

MLPB detector

- Shows the best BER performance over full range of interference.
- Difficult to implement in a hardwired structure for the time-varying statistic of received signal required by the detector.

References

1. Stok, A., Sargent, E.H.: Lighting the local area: Optical code-division multiple access and quality of service provisioning, IEEE Network **14** (2000) 42–46
2. Desai, B.N., Frigo, N.J., Reichmann, A.S.K.C., Iannone, P.P., Roman, R.S.: An optical implementation of a packet-based (Ethernet) MAC in a WDM passive optical network overlay, In: Proceedings of OFC. Anaheim, CA, USA **3** (2001) 1–3
3. Foschini, G.J., Vannucci, G.: Using spread-spectrum in a high-capacity fiber-optic local network. Journal of Lightwave Technology **6** (1988) 370–379
4. Foschini, G.J.: Time dynamics of outage of fiber optic local system employing spread spectrum. Journal of Lightwave Technology **7** (1989) 640–650
5. Kwong, W.C., Prucnal, P.R., Perrier, P.A.: Synchronous versus asynchronous CDMA for fiber-optic LANs using optical signal processing. In: Proceedings of IEEE GLOBECOM, Dallas, TX, USA, **2** (1989) 1012–1016
6. Salehi, J.A.: Code division multiple access techniques in optical fiber networks – Part I: Fundamental principles. IEEE Transactions on Communications **37** (1989) 824–833
7. Shrikhande, K., Srivatsa, A., White, I.M., Rogge, M.S., Wonglumsom, D., Gemelos, S.M., Kazovksy, L.G.: CSMA/CA MAC protocols for IP-HORNET: An IP over WDM metropolitan area ring network. In: Proceedings of GLOBECOM. San Francisco, CA, USA, **2** (2000) 1303–1307
8. Chang, F.T.W., Sargent, E.H.: Optical CDMA using 2-D codes: The optimal single-user detector. IEEE Communications Letters **5** (2001) 169–171
9. Shivaleela, E.S., Sivarajan, K.N., Selvarajan, A.: Design of a new family of two-dimensional codes for fiber-optic CDMA networks. Journal of Lightwave Technology **16** (1998) 501–508
10. Tancevski, L., Andonovic, I., Tur, M., Budin, J.: Hybrid wavelength hopping/time spreading code division multiple access system. Proceeding of IEE Optoelectronics **143** (1996) 161–166
11. Yang, G.C., Kwong, W.C.: Two-dimensional spatial signature patterns. IEEE Transactions on Communications **44** (1996) 184–191
12. Yang, G.C., Kwong, W.C.: Performance comparison of multiwavelength CDMA and WDMA+CDMA for fiber-optic networks. IEEE Transactions on Communications **45** (1997) 1426–1434
13. Chen, L.R.: Flexible fiber Bragg grating encoder/decoder for hybrid wavelength-time optical CDMA. IEEE Photonics Technology Letters **13** (2001) 1233–1235
14. Yegnanarayanan, S., Bhushan, A.S., Jalali, B.: Fast wavelength-hopping time-spreading encoding/decoding for optical CDMA. IEEE Photonics Technology Letters **12** (2000) 573–575
15. Yu, H., Breslau, L., Shenker, S.: A scalable web cache consistency architecture. In: Proceedings of ACM SIGCOMM '99 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. (1999) 163–174
16. Salehi, J.A.: Emerging optical code-division multiple access communication systems. IEEE Network **3** (1989) 31–39
17. Chang, T.W.F.: Optical Code-Multiple Access Networks: Quantifying and Achieving the Ultimate Performance. Master's thesis, Graduate Department of Electrical and Computer Engineering, University of Toronto (2000)

A Tool Measuring Operating System Supports for Squid Proxy Cache

Yuncheol Baek

Division of Software Science, Sangmyung University,
7 Hongji-Dong, Chongno-Gu, Seoul 110-743, Korea
bycker@sangmyung.ac.kr

Abstract. Unix-compliant open source operating systems are often used as platforms of web service systems. Choosing an operating system can affect web system performance. In this paper, we present a source-code-level time measurement tool and measure a service time of each system call that is invoked by Squid proxy cache. We produce the log while running Squid 2.4.STABLE1 on Linux 2.4.2 and Solaris 8. As a result, we find that Linux 2.4.2 provides better services than Solaris 8. This can be used as a guide for selecting system software on building web service.

1 Introduction

World-wide-web(WWW) is the most popular Internet service. There are so many web servers and web proxy cache softwares and they are divided into the open source software category and the proprietary software category. Representatives of open source software side are Apache[1] and Squid[2], leaders of commercial software side are Microsoft IIS[3], Inktomi[4], Cisco[5]. According to the surveys of Netcraft[6], above 58% of 38,444,856 web sites in the Internet world are using Apache (Table 1).

Table 1. A survey of web server software usage on Internet in February 2002

developer	Jan 2002	percent	Feb 2002	percent	change
Apache	20866868	56.87	22462777	58.43	+1.56
Microsoft	11097667	30.25	11198727	29.13	-1.12
iPlanet	1318991	3.60	1123701	2.92	-0.68
Zeus	792802	2.16	837968	2.18	+0.02

In the case of web proxy cache, Squid is the most widely used open source software. These kinds of open source web system softwares are usually running on the UNIX-compliant operating systems, specially open source operating systems like Linux and Solaris. In this paper, we describe a source-code-level time measurement tool. We apply this tool to Squid and analyze the operating system performance supporting Squid. Squid web proxy cache uses system calls frequently. It uses system calls of network subsystem for making connection with

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 239–247, 2002.

web servers and web clients. It also uses the system calls of the file subsystem for retrieving and storing the web objects from or to the disk cache. So, Squid is suitable to give a real intensive workload of web environment to an operating system. Our experiment was made as follows.

– Make a tool for measuring run time of program module.
– Patch Squid source code with APIs of our time measurement tool.
– Apply a workload generated by Web Polygraph[7] to Squid.
– Calculate elapsed time of each system call from the log generated by our tool.

The rest of this paper is organized as follows. In section 2, we introduce the related works of our experiments. In section 3, we describe the Squid's behavior associated with operating system services. In section 4, we describe our service time measurement tool. Result and analysis of this experiment are discussed in section 5. Conclusion is given in section 6.

2 Related Works

Many researches are presented on the performance of the web system softwares - web servers and web proxy caches. Banga and Druschel described the method of generating more realistic HTTP requests for measuring the capacity of a web server[8]. Barford and Covella made SURGE - a workload generator and evaluated the performance of network and server according as the load increase[9]. Web Polygraph mentioned in section 1 is the representative HTTP traffic generator for measuring responsiveness of a web proxy cache. Web proxy cache like Squid accepts clients' requests and get the very object from the other web servers if it is uncached. Next, Squid stores it into its local cache and supplies it to the client which issued the request. So, Web polygraph acts as not only clients generating HTTP requests to Squid but also remote servers storing web object that Squid needs. Maltzahn et al. measured the performance of enterprise level web proxies[11], Rousskov presented a good and detailed performance study[10] and Kim et al. gave a model for retrieval latency for proxy cache[12].

Cache replacement policy is the main topic of the web proxy cache research. Many studies are conducted for this area. One part of them carried out comparative performance study of existing policies and the others suggested new policies and evaluated them. The performance of the replacement policies of Squid was compared well in [13].

There are some works focused on the relations between web system software and operating system which is the basis of the web system software's running. Nahum et al. examined the several tuning methods for getting high performance of web service like using efficient socket function, reducing the number of memory copies in the system and minimizing the connection overhead under the TCP/IP protocols[14]. Mogul et al. suggested the operating system level supports for faster network servers[15]. Baek et al. mentioned the buffer cache mechanism of the UNIX file system is not adequate for the web proxy cache[16]. Our research

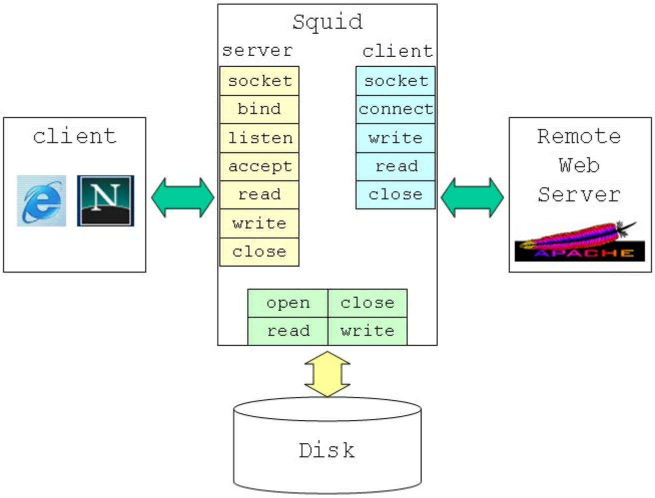


Fig. 1. System calls for Squid working

also comes into this category of investigating the correlations between the web proxy cache server and its underlying operating systems.

3 Flow of the Squid Working

Squid is derived from the ARPA-funded Harvest project. It is a high-performance proxy caching server for web clients, supporting FTP, gopher, and HTTP data objects. Unlike traditional caching software, Squid handles all requests in a single, non-blocking, I/O-driven process[2]. It is designed to run on Unix-compliant operating systems and it is free, open-source software. Many organizations and individuals use this software and IRCache Project of NLANR(National Laboratory for Applied Network Research) also adopts Squid for organizing US nation-wide cache hierarchy[17]. For the first step of our study, we analyzed Squid 2.4.STABLE1 source code and figured out the working flow of it. Fig. 1 summarizes the system calls that Squid makes during executing its operation.

As we can see in Fig. 1, Squid web proxy cache acts not only as a web server for the clients' request but also as a client to the remote server. Brief algorithm of Squid working is described as follows.

```
squid(){
    while(1){
        accept request of a client;
        if (the requested object is in cache){
            read object from the cache;
            send it to the client;
        }
        else {
            send request to the remote web server;
            if (the requested object is in remote server){
                read the object from remote server;
                write the object to the cache;
                send it to the client;
            }
            else
                send MISS reply to the client;
        }
    }
}
```

We can easily associate each line with the system call actually invoked except *send*. *Send* appeared in above algorithm means *write()* on network subsystem.

4 A Time Measurement Tool

We tried to measure the operating systems service time of the network-related system calls and file-related system calls while Squid is running. Squid consists of only one process and use 1024 *file descriptors* for network communications and file I/Os. It checks the status of each file descriptor periodically by polling. In the process of file operations, Squid doesn't read (or write) the whole size of the object. It reads (or writes) a file incrementally by the size of 4096 bytes each time. This mechanism prevents that some request holds the exclusive possession of Squid's service. Because of this design principle of Squid, we use the following data structures to accumulate the system call service time for each file descriptor in our time measurement tool.

```
struct PerFile{
    int fd;
    char *desc;
    char *name;
    struct timeval open;
    struct timeval read;
    struct timeval write;
    struct timeval close;
} SpendTime[1024];
```

We could get the elapsed time of each system call by stamping a time before and after the execution of a system call and calculate the difference of two times. The following two APIs perform this role.

```
void setStartTime(int fd, char *desc, char *name)
void setEndTime(int fd, char *desc, char *name)
```

API `setStartTime()` registers the present system time before the execution of file-related system calls. First parameter `int fd` is the index of a file descriptor,

Table 2. Hardware specifications

CPU	800Mhz Intel Pentium III Dual
Cache Memory	256KB
Main Memory	512MB
HDD	30GB SCSI
NIC	Ethernet 100Mbps

Table 3. Software specifications

OS	Solaris 8 Intel Platform Linux Kernel 2.4.2
Proxy Cache Server	Squid 2.4.STABLE1
Traffic Generator	Web Polygraph 2.7.3

second parameter `char *desc` is an absolute path of cached file, third parameter `char *name` is the name of a system call. `setStartTime()` and all other our time measuring APIs used C standard library function `gettimeofday()`. After the file-related system calls, `setEndTime()` checks the present system time for calculating the difference from the time stored by `setStartTime()` and stores the elapsed time to the array `SpendTime[]` according to the class of system calls. If the third parameter of `setEndTime()` is *close*, log is made out of the time in `SpendTime[]`. For measuring the service time of network-related system calls, the following two APIs are used.

```
void setStartSystemCallTime(void)
void calcSystemCallTime(char *name);
```

Being different from the case of the file-related system calls, it is not necessary to accumulate the each elapsed time of network-related system calls such as *socket(), bind(), listen(), accept(), close(), connect()*, because they could not be interrupted while their executions. API `setStartSystemCallTime()` gets the present time before system call and `calcSystemCallTime()` gets a difference from start time and put it into the log.

5 Experimental Results and Analysis

Hardware and Software usages of our experiment are listed in Table 2 and 3.

We used two kinds of operating systems, Solaris 8 and Linux Kernel 2.4.2. On top of each operating system, Squid 2.4.STABLE1 which is patched with our time measurement APIs were running. On the other system, traffic generator, Web Polygraph was running. We used the *PolyMix#1* which is the workload used for the first Cache-off[7] in 1999. It has the following characteristics, the reply size is chosen from an exponential distribution with a mean of 13Kbytes, 20% of the Polygraph client requests are uncacheable, latency is selected from a normal distribution with a 3 second mean and a 1.5 second standard deviation for each request, cache hit ratio of the requests is constant 55%. We made

Table 4. The service time of each system call for the case of 5 requests per second

		Solaris 8			Linux 2.4.2		
		number of calls	total service time (μsec)	ave service time (μsec)	number of calls	total service time (μsec)	ave service time (μsec)
file	open	12464.6	12747540.0	1022.7	12203.0	1154222.4	94.6
	close	12464.6	724651.0	58.1	12202.6	148770.2	12.2
	read	7868.2	2823406.2	358.8	7567.0	976454.8	129.0
	write	4539.6	3487230.2	768.2	4558.2	1121295.2	245.9
network	socket	8167.4	1079798.8	132.2	8227.0	212842.8	25.9
	close	26247.8	3878308.0	147.8	26302.6	841808.6	32.0
	bind (server)	2.0	138.8	69.4	2.0	39.2	19.6
	listen (server)	1.0	58.6	58.6	1.0	32.2	32.2
	accept (server)	18099.0	5203174.2	287.5	18098.6	593375.8	32.8
	read (server)	18099.0	1664708.0	91.9	18097.4	303329.2	16.8
	connect (client)	8179.2	1054448.6	128.9	8248.8	386326.6	46.8
	read (client)	41729.6	2835982.0	67.9	47107.2	1003498.4	21.3
	write	95190.0	5843834.6	61.4	96420.6	2884864.6	29.9

three kinds of experiments for varying the request rate - 5, 10, 15 requests per second. According to the empirical study of Rousskov[10], root-level cache in the cache hierarchy receives 15-20 requests per second at the peak time. 8-12 requests are given to Web proxy cache of *ISP*(Internet Service Provider) in a second. Leaf-level cache of an organization accepts 5-7 requests per second. So, selected numbers of 5, 10, 15 requests per second are sufficiently meaningful. Our experiments are carried out for 6 kinds of combinations - 2 types of operating systems and 3 categories of request rates. We repeated 1-hour long experiment 5 times, and took the mean value of the produced data. Table 4, 5 and 6 contain full details of the results.

For clear comparison, we can simplify the results. If Squid replies with cached object sized 13Kbyte (the mean value of the exponential distribution of reply size), Squid invokes following system calls in order.

- *socket()* for open a connection
- *accept()* a client's request
- *read()* URL from the client's request
- *open()* cached file
- *read()* cached file incrementally 4 times
- *write()* to the client
- *close()* the file
- *close()* the connection

We notice that file read() system call is invoked 4 times. It is because file I/Os are performed by 4096 Byte size at a time, so $\lceil 13312/4096 \rceil = 4$ file system read operations are needed. We can calculate the total time of system calls by

Table 5. The service time of each system call for the case of 10 requests per second

		Solaris 8			Linux 2.4.2		
		number of calls	total service time (μsec)	ave service time (μsec)	number of calls	total service time (μsec)	ave service time (μsec)
file	open	26180.8	52452389.0	2003.5	26057.6	2729217.2	104.7
	close	26180.8	1701902.8	65.0	26057.6	343152.2	13.2
	read	15251.8	4357569.4	285.7	14473.8	1812975.8	125.3
	write	10868.4	14290981.4	1314.9	11445.6	2964509.4	259.0
network	socket	18151.4	2508377.8	138.2	18760.2	541337.8	28.9
	close	54565.2	8185024.4	150.0	55019.6	1828493.8	33.2
	bind (server)	2.0	139.2	69.6	2.0	38.2	19.1
	listen (server)	1.0	57.6	57.6	1.0	31.6	31.6
	accept (server)	36450.0	10504843.4	288.2	36327.8	1205065.4	33.1
	read (server)	36449.6	3246299.2	89.0	36355.0	595169.4	16.3
	connect (client)	18183.0	2483005.0	136.6	18929.6	922424.2	48.7
	read (client)	95467.8	6824033.0	71.5	117688.2	2694690.8	22.9
	write	196503.4	12548965.4	63.9	206616.2	6527620.2	31.6

Table 6. The service time of each system call for the case of 15 requests per second

		Solaris 8			Linux 2.4.2		
		number of calls	total service time (μsec)	ave service time (μsec)	number of calls	total service time (μsec)	ave service time (μsec)
file	open	40088.2	112193710.8	2798.7	39968.2	15369979.0	384.6
	close	40088.2	2510847.0	62.6	39968.2	573839.0	14.4
	read	19423.8	4988762.4	256.8	18328.8	2449426.8	133.6
	write	20603.4	22088487.0	1072.1	21450.0	5921519.6	275.6
network	socket	31429.2	3666936.4	116.7	32314.0	1010803.2	31.3
	close	85650.0	10705142.8	124.9	86538.4	3012187.2	34.8
	bind (server)	2.0	109.4	54.7	2.0	39.2	19.6
	listen (server)	1.0	44.4	44.4	1.0	31.8	31.8
	accept (server)	54268.2	12711839.4	234.2	54287.0	1884169.6	34.7
	read (server)	54268.2	3661444.4	67.5	54283.0	921370.2	16.9
	connect (client)	31482.0	3616227.4	114.9	32364.8	1607084.6	49.6
	read (client)	174259.6	10914746.4	62.6	219559.2	5545847.2	25.3
	write	308611.4	17696386.0	57.3	332356.6	11434832.4	34.4

adding above latencies of system calls (Table 7). Similarly, the service time of the system calls for 13Kbyte uncached object can be calculated (Table 8).

In Table 7, Solaris 8 takes 4-5 times more service time than Linux 2.4.2. In Table 8 the total latencies of system calls of Solaris 8 is also 4-6 times larger than Linux 2.4.2. These results show that the system supports for Squid web proxy cache of Linux 2.4.2 is better than that of Solaris 8 in general.

Table 7. Total service time of system calls for the 13Kbyte size cached object(μsec)

	5 req/sec	10 req/sec	15 req/sec
Solaris 8	3237.0	3940.6	4489.4
Linux 2.4.2	760.3	762.2	1085.7

Table 8. Total service time of system calls for the 13Kbyte size uncached object(μsec)

	5 req/sec	10 req/sec	15 req/sec
Solaris 8	5412.6	8617.5	8226.8
Linux 2.4.2	1383.9	1462.5	1829.1

6 Conclusion

In this paper, we presented the design and implementation of a source-code-level time measurement tool in order to compare the operating systems supports for Squid web proxy server. We patched Squid 2.4.STABLE1 with our tool and gave a workload generated by Web Polygraph. As a result, Linux 2.4.2 is more suitable for running Squid than Solaris 8 in terms of the service time. We think that Solaris 8 have to be improved on their latencies of the system calls of file subsystem and network subsystem. This result can be used as guidelines for building web service with free software and for designing operating systems to support web services efficiently.

References

1. The Apache Software Foundation. http://www.apache.org
2. Squid Web Proxy Cache. http://www.squid-cache.org
3. Microsoft Corporation. http://www.microsoft.com
4. Inktomi Corporation. http://www.inktomi.com
5. Cisco Systems, Inc. http://www.cisco.com
6. Netcraft. http://www.netcraft.com/survey
7. Web Polygraph. http://www.web-polygraph.org
8. Banga, G., Druschel, P.: Measuring the Capacity of a Web Server. Proceedings of the USENIX Symposium on Internet Technologies and Systems. Monterey California (1997)
9. Barford, P., Crovella, M.: Generating Representative Web Workloads for Network and Server Performance Evaluation. Proceedings of the joint international conference on Measurement and modeling of computer systems. Madison (1998)151–160
10. Rousskov, A.: A Performance Study of the Squid Proxy on HTTP/1.0. WWW Journal 99. (1999)
11. Maltzahn, C., Richardson, K., Grunwald, D.: Performance issues of enterprise level web proxies. Proceedings of ACM SIGMETRICS international conference on Measurement and modeling of computer systems. Seattle (1997)
12. Kim, J., Bahn, H., Koh, K., Baek, Y.: Modeling of retrieval latency for proxy cache simulation. Electronics Letters, **37**, IEE, (2001) 167–169
13. Dilley, J., Arlitt, M.: Improving Proxy Cache Performance:Analysis of Three Replacement Policies. Internet Computing. **3**, IEEE (1999)

14. Nahum, E., Barzilai, T., Kandlur, D.: Performance Issues in WWW Servers. Proceedings of the international conference on Measurement an Modeling of Computer Systems. (1999) 216–217
15. Banga, G., Druschel, P., Mogul, J.: Better Operating System features for faster network servers. Proceeding of the Workshop on Internet Server Performance. (1998)
16. Baek, Y., Son, S., Koh, K.: Analysis on the UNIX Buffer Cache Mechanism of World Wide Web Proxy Server. Proceedings of the 16th IASTED International Conference Applied Informatics. Garmisch-Partenkirchen, Germany. (1998)
17. IRCache Project. http://www.ircache.net

The Design and Implementation of an Object-Oriented Process Control Loop Framework

Taewoong Jeon[1], Sunghwan Roh[1], Hyonwoo Seung[2], and Sungyoung Lee[3]

[1] Dept. Computer & Information Science, Korea University, Korea
{jeon,shroh}@selab.korea.ac.kr

[2] Dept. Computer Science, Seoul Women's University, Korea
hwseung@swu.ac.kr

[3] Dept. Computer Engineering, Kyunghee University, Korea
sylee@oslab.kyunghee.ac.kr

Abstract. Control loop is an essential part of the process control system that must control physical processes in which it is difficult or impossible to compute correct output value with input values alone. In this paper, we describe the design and implementation of a highly reusable object-oriented control loop framework to support the efficient development of real time process control applications. The basic building block in our control loop framework is the Point class. The Point class encapsulates process variables of a control loop together with control algorithms so that it can be easily adapted and extended to process control applications that have various structures and behaviors. The core of this paper is the design pattern of event/time-triggered Point class that can be used for flexible implementation of monitor and control functions required of target processes through the interaction of point objects performing continuous re-computation.

1 Introduction

The process control system produces the desired output from the input resources of a process responding adequately to the constantly changing environment, or continuously monitors and controls a process in order to maintain required relations among the objects in the process. In such a system, it is difficult or impossible to compute the correct output value with input values alone, and the time constraint is usually accompanied. A control loop is an essential part of the process control system [1].

An object-oriented framework provides an architecture that can be commonly used for the development of application programs that belong to a specific area or functional group. The architecture includes classes that can be easily adapted and extended. Accordingly, a complete application system can be easily built by modifying some of the classes in a framework and appending additional functions to the framework, then connecting them to the building blocks of the framework [2].

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 249–261, 2002.

On the other hand, much research has been done recently for the development of real-time systems based on the object-oriented methodology. S. Faulk et al. in [3], tried to make requirement specifications in the real-time software development mathematically more rigid and easier to share, by putting object-orientation, graphic representation and standardized approaches together. B. Selic et al., in [4], provided more accurate and simple system modeling strategies by graphically representing the real-time system architecture through object-oriented methodologies. The system model, practicable at all the levels of abstraction, helps find out requirement or design drawbacks at the early stage of development.

While the development of general real-time systems is treated in [3] and [4], this paper focuses on the development of a process control framework with higher reusability. That is, the purpose of this paper is to make development of a process control system more efficient and easier by studying the methods to design common parts of process control application programs as an object-oriented framework with higher reusability and flexibility. In such a framework, the structure of a control loop can not only be modified dynamically during the execution of an application, but it can also be easily extended to a single control loop with many process variables or a complicated control loop application with many single control loops. The core of this paper is the design pattern of the Point class which flexibly supports the continuous re-computation of process variables of a control loop. A control loop framework is composed of frozen parts and hot spots. The frozen parts implement common functions of various control loops and can be used without modification during an application development. The hot spots represent the variability between control loops. A control loop application can be completed by adapting and extending the hot spots of the framework according to the system requirements, and then compounding them to the frozen parts of the framework.

This paper is organized as follows. Sect. 2 analyzes the domain of the control loop framework to be developed, explains the control loop model, and reviews requirements to be considered in order to develop a highly reusable and flexible framework. In Sect. 3, we propose a control loop framework designed to be easily adapted and extended to process control application systems with various structures and behaviors to meet the requirements presented in Sect. 2. In Sect. 4, an example of implementing a control loop application using the proposed control loop framework is explained. Sect. 5 reviews previous researches on the process control software designed as an object-oriented framework. Sect. 6 presents our conclusion and future work.

2 Domain Analysis of the Control Loop System

2.1 Control Loop Model

A process control system consists of a process and a control which controls and monitors the process. The current status of the process can be represented by process variables. There are four types of process variables: controlled variable,

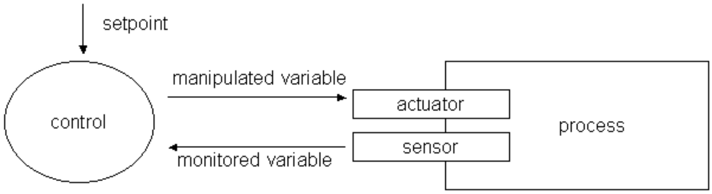


Fig. 1. The Control Loop Model

input variable, manipulated variable and reference variable. A controlled variable represents an actual measured value of an object to control. An input variable, not an object to control, is a process variable which represents an input value needed to control the process. Both variables are used as process variables monitored by the control. That is, they are monitored variables of the process. A manipulated variable can be directly modified by the control. A reference variable represents a setpoint, that is, a target value of the controlled variable [1].

Since the external environment of a process control system is indeterministic, unstable and constantly fluctuating regardless of the process, it is difficult or impossible to determine the correct output value with values of input and internal state alone. Therefore, in order to meet or maintain the requirements of the process outputs specified by the reference variables, the control computes the values of manipulated variables based on the reference and monitored variables and sends them to the process. The process, in turn, controls the monitored variables according to the values of the manipulated variables entered from the control. Such a process control system can be regarded as a control loop model (Fig. 1). A control loop model is composed of two components, a control and a process, and two connectors which provide paths of the asymmetric and cyclic flow of data.

2.2 Requirements of the Control Loop Framework

Following are the requirements considered in this paper to enhance reusability and flexibility of the control loop framework.

- By adapting and extending hot spots of the control loop framework, it should be possible to easily build up a control loop application which implements interactions of process variables and control algorithms required in a specific system. It should be also possible to dynamically modify the configuration of the control loop during execution of the application.
- It should be possible to extend to a complex process control application which includes many control loops, not just one.
- The control loop application should be both time-triggered and event-triggered.
- Simulations of the control loop application should be possible, and once they are done, the application should be easily migrated to the real process control environment.

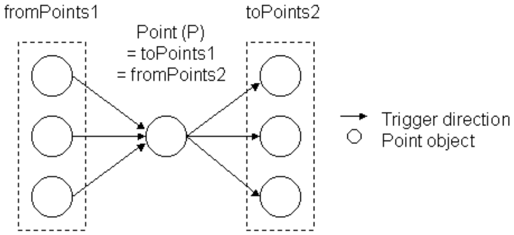


Fig. 2. Relation between the Point P and fromPoints/toPoints

- If the user of the framework wants alarming or watching function, it should be added with ease.

3 Design of the Control Loop Framework

3.1 Point Class

The control loop controls the process through an actuator so that the state of the process can be kept within the setpoint, and monitors the state of the control through a sensor. The control loop system timely reacts to the value changes of the monitored variables entered from the sensor and recomputes the values of the manipulated variables. Then, it sends the result to the actuator to change the values of the controlled variables of the process. The control loop framework proposed in this paper is designed to have the Point class as its basic building block which encapsulates process variables together with control algorithms in order to flexibly constitute the interrelationship among the process variables.

A point can either be an input point or a computed point according to the way its value is determined. While the value of an input point comes from an external data source, a computed point gets its value from the other points inside the system. That is, a computed point depends on input points or other computed points to determine its value, and registers itself as a destination point(toPoints) to those points. If the value of a point changes, it notifies all the registered toPoints of the change. A toPoints, then, reads the values of all the source points(fromPoints) to recompute its value using its own formula. Continuous re-computation is performed as the change of point values is propagated to all the toPoints connected through the dependency chain. Fig. 2 shows the relation between a point P and fromPoints/toPoints. The point P becomes toPoints1 from the viewpoint of fromPoints1, and fromPoints2 from the viewpoint of toPoints2.

Loose coupling among the objects must be maintained in order to easily adapt and extend the process control framework to a process control application system that has various structures and behaviors. The point objects can be loosely coupled by representing control relationships among them using the Observer Pattern and the Composite Pattern [5] in design.

Fig. 3 shows the design pattern of the Point class and depicts the relationship among the Point class and its subclasses, InputPoint and ComputedPoint class.

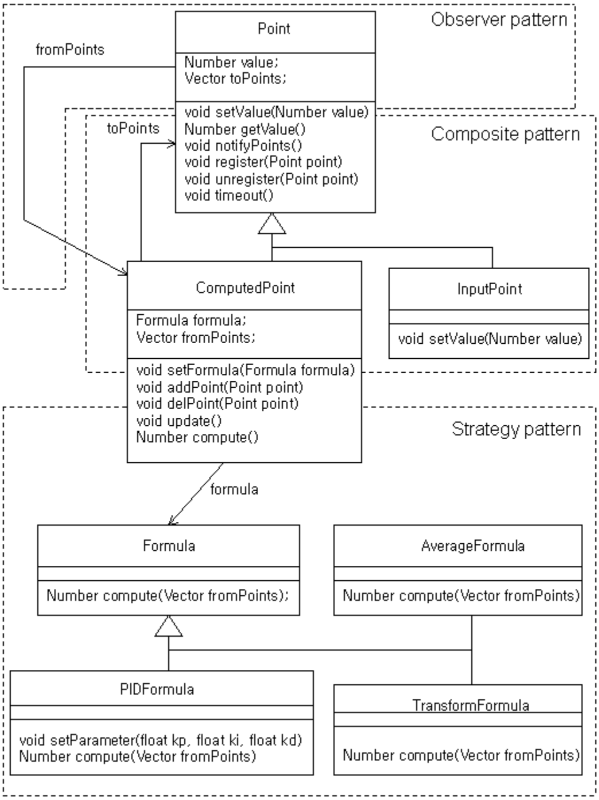

Observer pattern
Point
Number value;
Vector toPoints;
void setValue(Number value)
Number getValue()
void notifyPoints()
void register(Point point)
void unregister(Point point)
void timeout()
fromPoints
toPoints
Composite pattern
ComputedPoint
Formula formula;
Vector fromPoints;
void setFormula(Formula formula)
void addPoint(Point point)
void delPoint(Point point)
void update()
Number compute()
InputPoint
void setValue(Number value)
Strategy pattern
formula
Formula
Number compute(Vector fromPoints);
AverageFormula
Number compute(Vector fromPoints)
PIDFormula
void setParameter(float kp, float ki, float kd)
Number compute(Vector fromPoints)
TransformFormula
Number compute(Vector fromPoints)

Fig. 3. The Point Class

The ComputedPoint class responds to the value changes of the process variables in the Point class whose destination point(toPoints) is the Computed Point. It plays the role of observers. The ComputedPoint also creates a Composite pattern in which it has other point objects as its components through the fromPoints reference. The InputPoint gets its value directly from the outside through the setValue member function, while the ComputedPoint computes its value using the values of the other points affecting it. Therefore, the fromPoints reference in Fig. 2 can be either an InputPoint object or a ComputedPoint object, but the toPoints reference can be a ComputedPoint object only. In Fig. 4, when a value is set to anInputPoint, its notifyPoints member function is called, which, in turn, calls the update functions of all the toPoints of the InputPoint. After the update function of the invoked aComputedPoint1 re-computes its point value through the formula class(aFormula1), it calls the update functions of its toPoints(aComputedPoint2). Since values of the monitored variables entered from the sensor end up with starting the actuator through the chain reaction of the point re-computations and update calls, an expected control effect can be achieved.

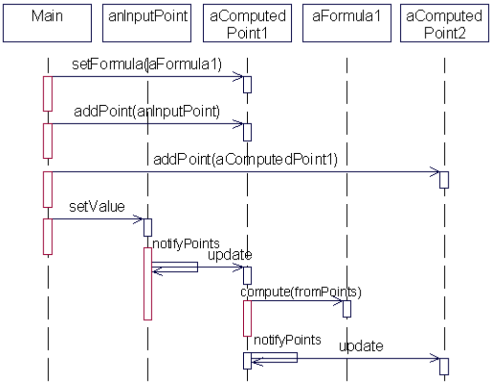


Fig. 4. Interactions of Point Objects

3.2 Formula Class

The manipulated variables, the monitored variables and the actuator can be regarded as computed points. Their structures are the same, but the behaviors of their formula functions are different. So, we applied the Strategy pattern [5] to encapsulate the formula in a separate class in designing a computed point.

The computation algorithm of a computed point is determined by a Formula class object which is dynamically replaceable according to the Strategy pattern. The computed point class has a formula which is a reference to the Formula class object. The Formula class provides a compute member function which is a common interface to the various algorithms a computed point may have. Using the compute function, a computed point invokes an algorithm defined in a subclass of the Formula class. If a computed point object represents a manipulated variable using the PID control algorithm, it references to the PID Formula. If a computed point object represents a monitored variable that contains an average value of many sensors, it references to the Average Formula class object. If a computed point object represents an actuator, it references to the Transform Formula. The compute member function of the PID Formula class contains PID control algorithm, and the setParameter member function reads parameter values required for the PID control algorithm when the program starts. By making the reconfiguration of parameters possible, the system can be easily modified without rebuilding the program.

3.3 Lazy Point Class

If a value of a certain computed point is used infrequently, it is a waste of time to re-compute its value every time the values of points it is dependent on change. It is more efficient to re-compute when the value is requested. In order to meet such a requirement, a LazyPoint class [8] is incorporated in this paper as a subclass of the ComputedPoint class. A LazyPoint re-computes its value only when it

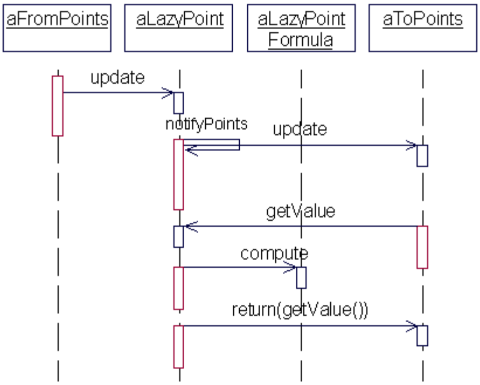


Fig. 5. Interaction of the LazyPoint

is used by another point, while a ComputedPoint is re-computed whenever the values of fromPoints change.

As shown in Fig. 5, a LazyPoint simply records an update and delays the re-computation of its value until it is requested, while a ComputedPoint re-computes its value every time its update function is called, and notifies the registered points of the change. Only when a record of an update exists, its value is re-computed, otherwise, the stored point value is sent. So, time waste for the unnecessary point re-computation can be avoided.

3.4 Time Triggered Point Class

In the control loop system, not only the event triggered approach, in which the system reacts to the state change of the process, but also the time triggered approach must be utilized, so that the system can reach or maintain the setpoint value of the process within a required time by monitoring the control state of the process at specific points in time regardless of the state change of the process. A Timer class is defined in the system to meet such requirement.

As explained above, the manipulated variable point of the control loop computes its value based on the values of the setpoint and the monitored variable, and sends the value to the actuator. If the value of the monitored variable does not reach the setpoint value after a specific period of time, it might be necessary to raise an alarm condition. To represent such a manipulated variable, the ManipulatedVariablePoint class which needs a timer is provided as a subclass of the ComputedPoint class. General manipulated variables which do not need alarming function are created from the ComputedPoint class.

The TimeTriggeredSensor is periodically activated by the Timer, and the EventTriggeredSensor is activated in case the value of the sensor changes. Both sensors are created by inheriting from the InputPoint. The Point objects that need a timer such as the time triggered sensors, manipulated variables with alarming functions, or the Simulation Point objects register themselves and the duration of time to the Timer. The Timer, at specific intervals of time, invokes the timeout member function in the registered Point object and notifies the registered time has passed. The manipulated variable whose registered time has passed checks if the monitored variable has reached the setpoint, and raises an alarm condition if it has not.

3.5 Simulated Process

The SimulationPoint class represents a simulated process by simulating the interaction between an actuator and a sensor. The effect caused by the actuator is sent to the SimulationPoint object, which, in turn, transfers the simulated value based on the effect to the sensor after a certain feedback delay time. Therefore, when an application which is made out of a framework is to run under the actual operational environment, the SimulationPoint is only to be replaced by the actual process. The starting point of the SimulationPoint is the actuator, and the destination point is the sensor. Once the actuator object invokes update function of the SimulationPoint object, the SimulationPoint object computes a simulated value through the SimulationFormula object it references to. When the Timer object of the SimulationPoint invokes timeout function after the feedback delay time passes, the simulated value is sent to the sensor.

3.6 Architecture of Entire Classes of the Control Loop Framework

Fig. 6 shows the inheritance and composition relationship among the classes of the control loop framework as a whole. Since the fundamental concept of the framework design in this paper is the continuous re-computation of point values, the various classes that consist of the control loop framework are inherited from the Point class.

4 An Application Using the Control Loop Framework

Using the control loop framework proposed in this paper, it is possible to design a flexible control loop software component based on the control loop model in Fig. 1. To implement a control loop application, the control loop system developer using the framework can build the classes his/her application requires by inheriting and extending the Point class or its subclasses. This section explains a control loop application implemented using the Java programming language.

Fig. 7 shows a heat control loop model which controls room temperature. The control is a heat control to maintain the desired room temperature as a setpoint, and the process is a heater monitored and controlled by the heat control. The monitored variable represents the current room temperature taken by the sensor. The temperature is a feedback value to the control and used to compute the value of the manipulated variable, which, in turn, is sent to the actuator.

Fig. 8 shows a configuration of the heat control loop in Fig. 7 constructed using the Point classes in the control loop framework. The arrows represents trigger directions between the connected Point objects. The setpoint and the sensor

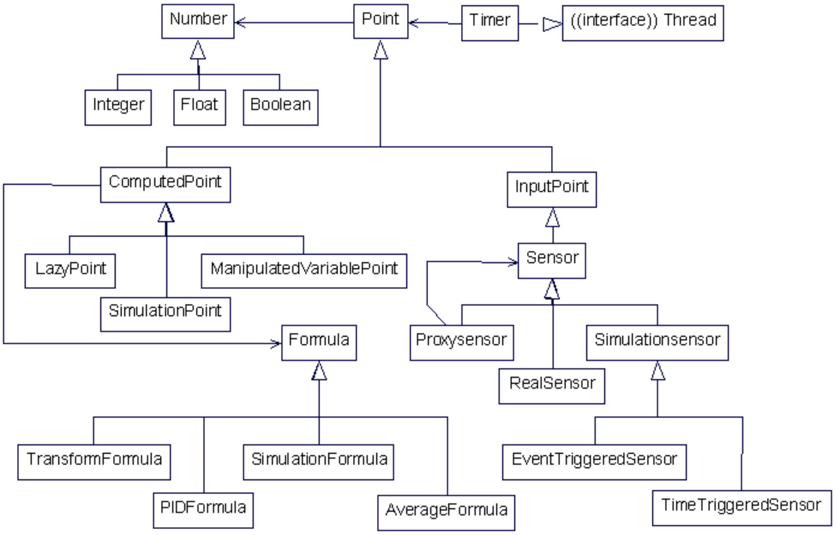


Fig. 6. The Control Loop Framework

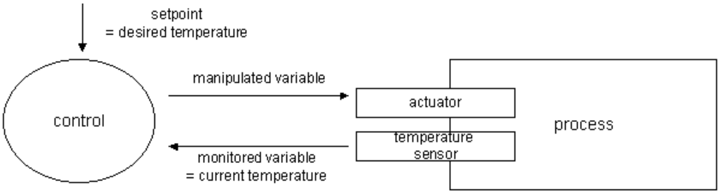


Fig. 7. The Control Loop Framework

are InputPoint objects of the framework. The manipulated variable, monitored variable, actuator and the simulation point are all ComputedPoint objects, and each references to its own Formula object.

The control mechanism of the heat control loop system is as follows: The sensor is triggered by the sensor timer at specific periods of time, and sends the measured temperature to the monitored variable. The manipulated variable takes the monitored value from the monitored variable and computes the manipulated value using the PID control algorithm of the PID formula. The computed manipulated value is sent to the actuator. Then, the value is reflected on the simulated process. The manipulated variable timer notifies the manipulated variable that a certain period of time has elapsed. Then, the manipulated variable, with the monitored value and the setpoint value, judges whether the control is in the normal state or not.

The value of the actuator is sent to th simulation point. As soon as the simulation timer notifies the simulation point that the feedback delay time has elapsed, the simulation point sends the current temperature to the sensor. The value of the sensor is, in turn, sent to the monitored variable to make a loop.

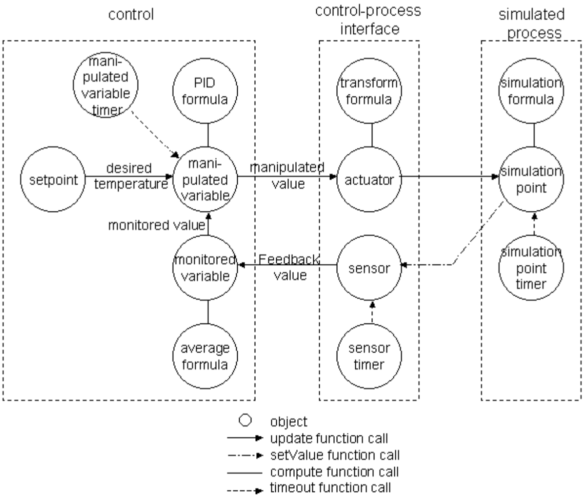


Fig. 8. The Heat Control Application

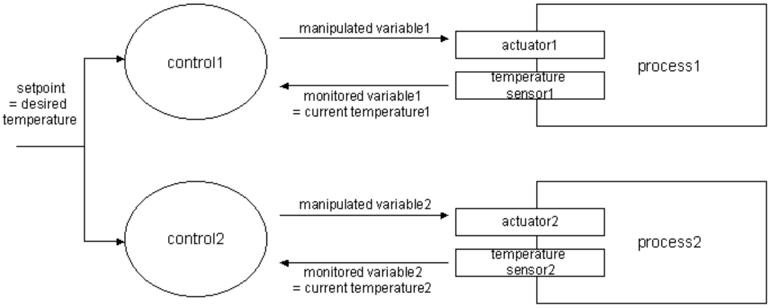


Fig. 9. A Control Loop System with Two Single Control Loops

Fig. 7 is a single control loop system where there is only one point object for each of the actuator, sensor, monitored variable and manipulated variable. There exists a single control loop system with many process variables, or a control loop system with many single control loops. Such complicated control loop systems can also be easily implemented using the control loop framework.

Fig. 9 shows a control loop system which consists of two single control loops. An application of such a system can be constructed using Point classes of the control loop framework as shown in Fig. 10.

5 Related Works

Current researches on the design of the process control software in an object-oriented framework are reviewed in this chapter.

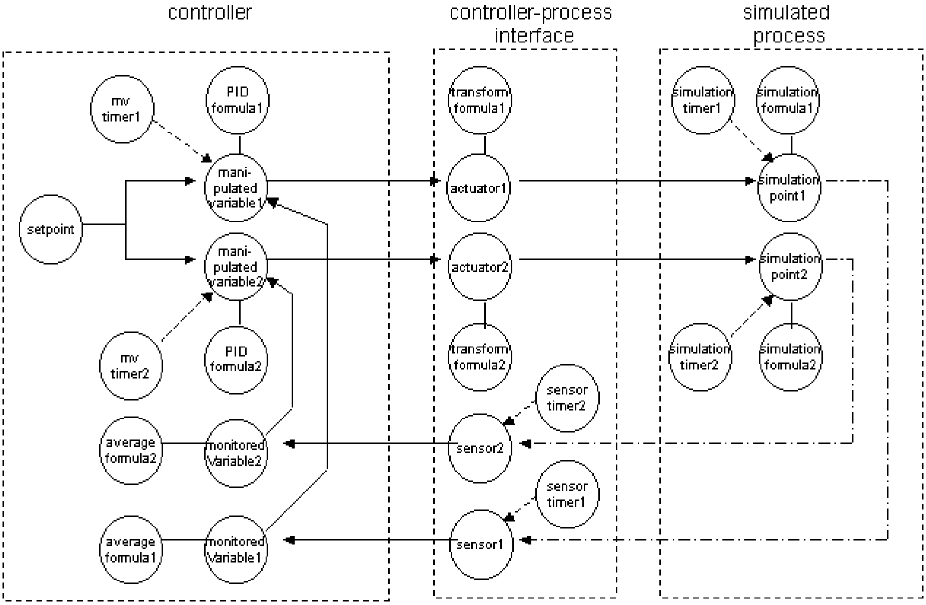


Fig. 10. An Application of the System in Fig. 9

Per Dagermo and Jonas Knutsson [8] regarded the process control as a continuous re-computation of a number of output values as the response to changes to a number of input values. A value in [8], which is an abstraction of a process variable, can either be an input value or a computed value. When a value changes, it notifies all the computed values registered as its dependents. When a dependent value is notified, it re-computes itself.

P. Molin and L. Ohlsson [9] designed a framework for fire alarm systems, where the logical behavior of the input/output devices such as sensors and actuators at the interface are standardized and defined as Points [9]. The concept of Point was applied in this paper as the Point class.

Jan Bosch [10] designed an object-oriented framework for the measurement system which measures the quality of manufactured products and picks out low-quality products. The Strategy pattern and the Composite Pattern were mainly utilized in the design of the system. The control loop of the measurement system differs from the one proposed in this paper in that the measured values are not used directly to control the manufacturing process. In this paper, the measured value, that is, the monitored variable is used as the controlled variable to control the process.

The process variables as well as sensors and actuators in this paper are all defined as Point classes or its subclasses performing continuous re-computation. While data types, computation algorithm, and control of the computation are all encapsulated in the Value class in [8], they are separated in different classes in this paper. As the data type of a process variable is separated from its control and defined in a Number class, the operations of the process variable are performed polymorphically. This allows the control loop to have multiple types for its point values. We defined computation algorithms in separate classes according to the Strategy pattern so that the algorithms used by computed points are easily replaceable at runtime.

6 Conclusions and Future Work

This paper has described the design of a highly reusable control loop framework for process control systems. The core of the proposed control loop framework is the design pattern of the Point class which encapsulates state values of the process together with the control mechanism. As subclasses of Point are designed using the Observer and Composite patterns, Point objects are loosely coupled with one another and can be easily reconfigured as needed during the execution. Application developers using the control loop framework can complete their applications efficiently by adapting and extending the Point (sub)classes to their requirements.

Most process control systems require parallel and real-time processing. While the proposed control loop framework utilizes Java's multi-thread functionality for parallel processing, it does not directly support real-time scheduling. The framework instead assumes the use of a real-time kernel when extended to an application. The primary work we plan to do in future is to enhance the control loop framework with real-time features to ease its extension into specific real-time operating system environments.

Acknowledgements

This work was partially supported by a Korea University Grant 2001, and by the Basic Research Program of the Korea Science & Engineering Foundation under Grant No. R01-1999-00238.

References

1. M. Shaw: Beyond Objects: A Software Design Paradigm Based on Process Control, ACM Software Engineering Notes, **20** (1995)
2. G. F. Rogers: Framework-Based Software Development in C++, Prentice Hall, (1997)
3. S. Faulk, J. Brackett, P. Ward, and J. Kirby Jr.: The Core Method for Real-Time Requirements, IEEE Software, (1992) 22–23
4. B. Selic, G. Gullekson, and, P. T. Ward: Real-Time Object-Oriented Modeling, John Wiley and Sons, (1994)
5. E. Gamma, et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, (1995)
6. S. Bennett: Real-time Computer Control: An Introduction, 2/e, Prentice Hall, (1994)

7. B. Woolf: The Abstract Class Pattern, Pattern Language of Program Design 4, Addison-Wesley, (2000)
8. Per Dagermo and J. Knutsson: Development of an Object-Oriented Framework for Vessel Control Systems, Technical Report, Dover Consortium (1996)
9. P. Molin and L. Ohlsson: Points & Deviations - A Pattern Language for Fire Alarm Systems, Pattern Languages of Program Design 3, Addison-Wesley, (1998)
10. J. Bosch: Design of Object-Oriented Framework for Measurement Systems", Object-Oriented Application Frameworks, John Wiley, (1998)
11. K. Arnold, J. Gosling, and D. Holmes: The Java Programming Language, 3rd edition, Addison-Wesley, (2000)

A Personal Internet Live-Broadcasting System*

Sangmoon Lee[1], Sinjun Kang[2], Byungseok Min[1], and Hagbae Kim[1]

[1] Department of Electrical and Electronic Engineering
Yonsei University, Seoul, Korea
hbkim@yonsei.ac.kr
[2] ACS Technology Co., Ltd., Seoul, Korea

Abstract. In the paper, we present an internet personal-live broadcasting server system. Our solution enables amateur users to broadcast with basic multimedia equipments. For the scalable system architecture, the broadcasting server is configured with a cluster of streaming units. For scalable broadcasting services, we build multiple-channel establishment and channel expansion. Concurrent services for a number of broadcasting channels are effectively provided, and also the capacity of channel can be expanded by connecting another group according as the number of participants increases. Furthermore, for the sake of complete live broadcasting with high quality of transmission, it supports both TCP and UDP according to status of network environments and received packet loss in the user system. The performance of the system is effectively evaluated at such practical commercial sites as well-known community and e-business sites.

1 Introduction

Recently, as internet environments are improved, a variety of applications to meet sophisticated requirements of users have been served in the internet. While general web contents provide static pictures or text, contemporary web pages generally consist of dynamic contents like CGI(Common Gateway Interface). Furthermore, as multimedia services become popular, demands for video and audio streams are accordingly increasing. Thus, a web server should be able to deal with heavier accesses and higher volume of data. Specially, in such commercial sites as community, e-business, VOD(Video On Demand), and broadcasting services, it should be able to guarantee reliable services as well as maintenances of high performance. There were previous works to improve the QoS(Quality of Service) for the multimedia streaming on the current best-effort internet with partial success to achieve these features[1,2]. The authors of [3,4] presented principal techniques and algorithms for such streaming video services over internet as broadcasting data compression, streaming server, and caching. In particular, with a view to guaranteeing QoS over multimedia servers, there were also proposed works that included resource-management mechanisms embedded in a middle-ware layer[5], an adaptive QoS server model to cope with a variety

* This work is done by the MOCIE project; the Development of High-Performance Scalable Web Servers.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 263–271, 2002.

of circumstances[6], and its implementation[7]. Additionally, for achieving reliable multimedia services that primarily handle continuous audio and video streams, a distributed structure of multimedia servers[8], a data placement and disk scheduling algorithm[9], and a scalable storage management[10] have been investigated as well. It is nevertheless difficult that these results are effectively applied to practical systems. In the paper, we present an efficient internet live-broadcasting server system. The system allows users to easily broadcast and to actively participate in the channel as well. Though various kinds of internet broadcastings are professionally serviced, our solution is not just for experts[11] but basically for amateur users to broadcast by equipping with such simple and basic audio/video equipments. In the broadcasting service, a fixed number of channels and its capacity usually restrict a scale of service. As the number of users is growing, those inflexible components result in not only degrading service qualities but also interrupting continuous services. We, thus, focus on scalability of channel establishment and capacity, where different channels can be readily established in streaming units so that broadcastings are concurrently provided for users. The capacity of a channel is easily expanded by simply connecting groups according as the number of participants increases. The architecture of the broadcasting server system is intrinsically based on a cluster of streaming units so as to construct highly scalable systems. The system performance is validated in the field test conducted at two commercial sites, practically adopting our solutions.

The paper is organized as follows. In Section 2, we present the basic architecture of our overall system and the detailed functions of its parts. Section 3 describes the flexible scalability of the broadcasting channels and the server system. Section 4 discusses the mode switching between TCP and UDP in order to efficiently transmit the broadcasting data. Section 5 validates the performance of our system through field tests, whereas the paper concludes with Section 6.

2 System Architecture

The basic environments for proper services are depicted in Fig. 1. We develop a server system and solutions for internet personal-live broadcasting. The server system is intrinsically composed of such three primary parts as a web server, the CSM(Cluster Server Manager) with backup, and clustered streaming units. The modules constructing each part are presented in Fig. 2. While the web server provides user applications to clients, the other parts operate transparently to users. The CSM controls a broadcasting and properly manages each of streaming units, and its backup server is prepared to guarantee the feature of High Availability(HA). The clustered streaming units are real broadcasting servers that are directly connected with user systems. The main functions of system modules in Fig. 2 are described as follows.

2.1 The Client Part and Web Server

We select the Apache that is widely adopted for the web servers. It naturally makes connection with users and broadcasting servers. When a user accesses

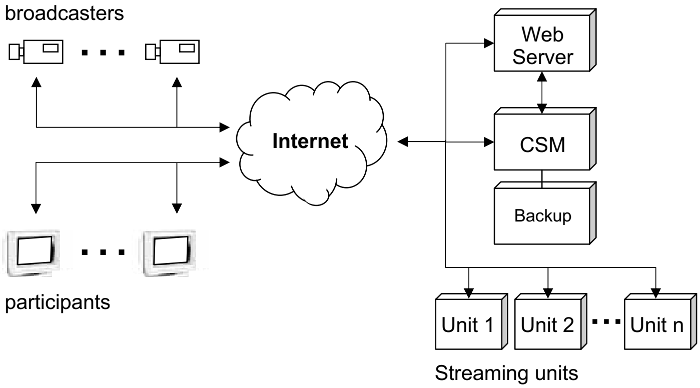


Fig. 1. Overview of the live-broadcasting system

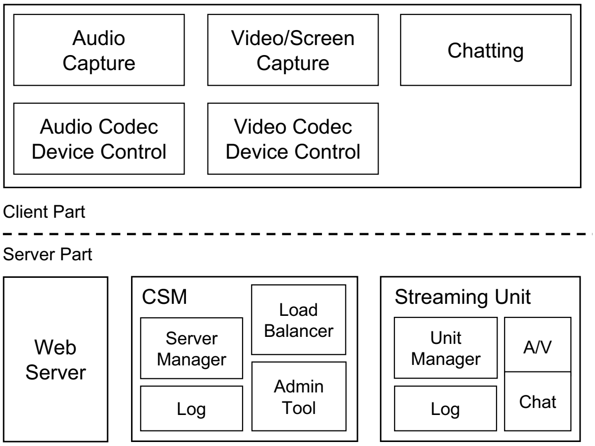


Fig. 2. Construction of system modules

the web server, it provides a user application in the form of ActiveX[12] with modules in the client part.

The user application shows current broadcasting status receiving from the CSM, and returns user requests such as creating channels or participating in a broadcasting. To transmit broadcasting data, it utilizes the capture and codec device control module for both audio and video data, and the chatting module for text data, respectively.

2.2 The CSM

The CSM basically consists of *log*, *server manager*, *load balancer*, and *admin. tool*. It plays a key role of managing broadcasting services as well as individual streaming units. Since it can be seriously a single failure point in clustered streaming unit, the backup server detects periodically with Heartbeat[13,14] to examine whether the CSM is dead. Each streaming unit reports such instant broadcasting information to the CSM as the channel number, the group number, and the number of users in each channel. This information is recorded in a log file, which is applied for establishing and releasing channels. Furthermore, the load balancer controls a workload among streaming units by appropriately allocating both the channels and the participants. The activity of each module in the CSM is summarized by:

Log: The broadcasting information is recoded.

Server Manager: It interfaces with service administration tool(web application) and reports channel status to the web server.

Load Balancer: It manages a maintaining proper balance of allocating channels and participants in each streaming unit.

Admin. Tool: Administration tool interfaces with the monitoring program for a system administrator.

2.3 The Streaming Unit

The streaming units are real broadcasting servers that practically work in the normal operation. The groups in the streaming unit are logical places at which both channels and participants are allocated, and to which such broadcasting data as audio, video and/or text data are sent. When a certain streaming unit fails in the cluster, the unit manager logically eliminates it from the cluster in order to prevent malfunctioning or propagating its adverse effects. The primary functions of the streaming unit are also described as follows.

Log: Events occurring in the streaming units are recorded.

Unit Manager: It controls an establishment and release of channels.

A/V: It transfers audio and video streaming.

Chat: It transfers text data.

3 Scalability of the Broadcasting Services

As the number of users significantly increase, the system eventually suffers from the overload decreasing the quality of broadcasting, and moreover the service may be discontinued. To overcome these problems and maintain the services without stopping, the scalability of channel establishment and capacity should be guaranteed. Our system is designed to maximize the number of the active channels and participants available within permitted system resources that increase linearly by adding streaming units into the cluster. The capacity of a channel can also be expanded by connecting groups. The CSM makes the logical connection with empty groups to prepare some redundancy before exhausting the channel capacity. Thus, a channel is made of groups from different streaming units to easily be expanded like Fig. 3.

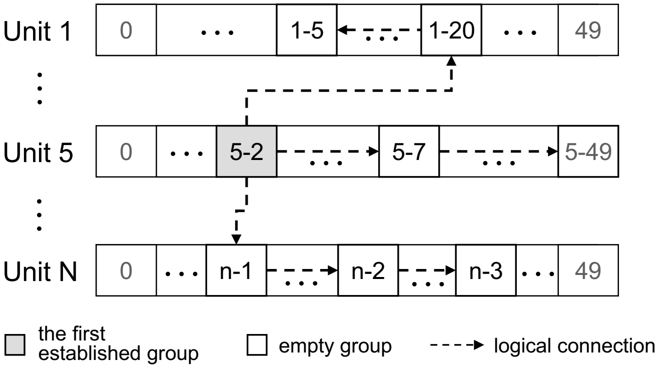


Fig. 3. Expansion of a group

For example, consider *N* streaming units that has 50 groups, each numerated from 0 to 49, as depicted in Fig. 3. The first established group 5-2 is expanded in the following scenario. The group 5-2 begins with being connected to the empty group in the same streaming unit(5-2 → 5-7 → 5-49). However, unless there is an empty group to connect, it turns to another streaming unit(5-2 → 1-20 → 1-5, 5-2 → n-1 → n-2 → n-3).

4 Broadcasting Mode Switching

A mode-switching scheme between TCP and UDP can be adopted appropriately for achieving particular purposes. To reduce traffic amount of web servers, the authors of [15] proposed a hybrid TCP-UDP transport according to the length of HTTP transmission time. Our system places importance on delivering both audio and video data completely to the participants under various network environments. Since UDP generally takes less overhead than TCP, the broadcasting data is transmitted using UDP at initial time. However, if a user system is located in the network that uses an internal IP, for example in the form of NAT(Network Address Translation), any data cannot be transmitted through UDP [3,16]. Additionally, packets loss due to heavy traffic may cause to significantly degrade the quality of the broadcasting service. Considering such a complete transmission failure or lower service quality in the user side, our system also supports to TCP mode for particular users as shown in Fig. 4. This is another form of a hybrid TCP-UDP mode. The channel established by a broadcaster is assigned to any available group, which readily transfers broadcasting data to participants. The data is stored continuously in the circular-typed buffer of the group. Fig. 5 is the logical structure of the buffer in each group. After being switched by TCP mode, the data blocks in the buffer are transmitted. When the data-transmission speed is faster than the storing rate, the currently-transmitted data block cannot exceed the storing point. However, in the lower transmission speed than storing, only audio data is serviced in order to avoid data update before sending.

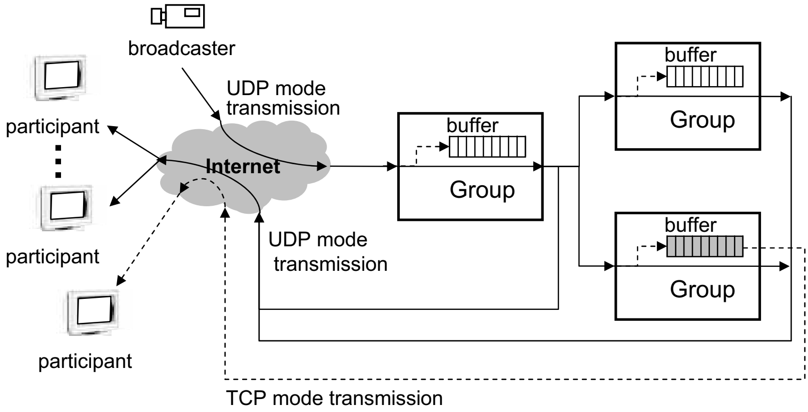


Fig. 4. Broadcasting data transmission using UDP and TCP

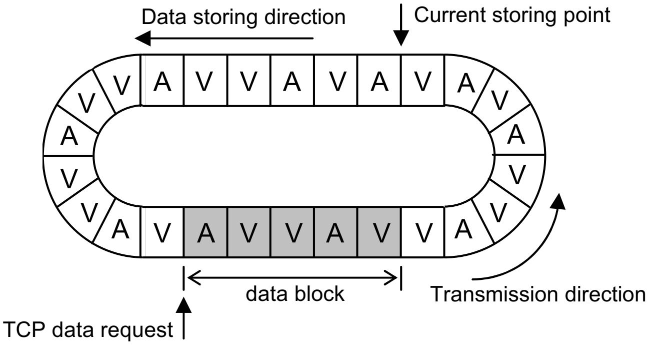
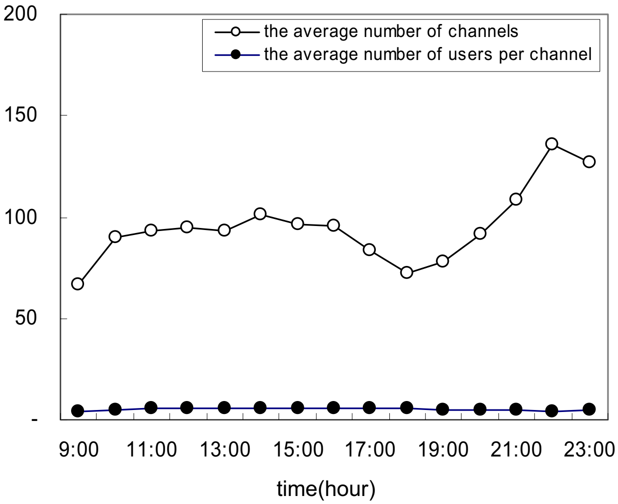


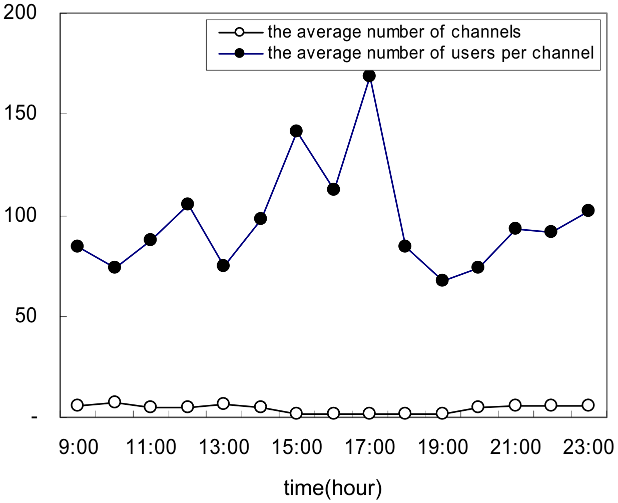
Fig. 5. Data transmission in the buffer (A: audio data, V: video data)

5 Analysis of System Performance

The performance of the system is effectively tested at two commercial sites that practically adopt our solutions. Two sites, LivePAX in the PAXNET[17] and Personal Broadcasting in the LYCOS KOREA[18], are well-known companies to provide active internet broadcastings. The former focuses on a stock-quotation broadcasting, whereas the latter provides personal broadcasting services. For each individual streaming unit, the average numbers of users and channels visiting for one day are given in the Fig. 6. These patterns are repeated daily because of particular characteristics of the services. It presents that multiple channels are established(Fig. 6(a)), on the opposite manner, a number of users are in a few expanded channels(Fig. 6(b)). Specially, Fig. 7 validates the solid scalability of the channels. Since the feature of service is like Fig. 6(b), a lot of active groups are working and forming each channels. Therefore, the scalability of the system is naturally confirmed for expanding the channel capacity as well as establishing multiple channels.



(a) LYCOS KOREA



(b) PAXNET

Fig. 6. Performance of the services in a streaming unit

6 Conclusion

We present a server system for internet personal-live broadcasting. It is not just for experts but also for amateur users who want broadcasting with basic multimedia equipments. For highly-scalable broadcasting services, we develop a server system with proper broadcasting solutions. The capacity of each channel is easily expanded by connecting groups according as the number of participants

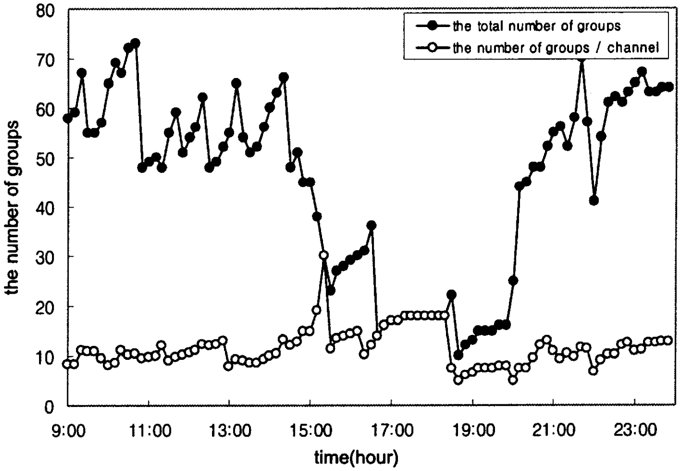


Fig. 7. Scalability of the channel in PAXNET

increases. The multiple channels can also be established on the same streaming units, so that it can provide concurrently a number of broadcasting channels to users. In addition, considering various network environments of user systems and qualities of the broadcasting data, the system supports transmission-mode switching between TCP and UDP. The effective performance of the system based on states of practical well-known websites are validated through field tests.

References

1. Furht, B., Westwater, R., Ice, J.: Multimedia broadcasting over the Internet. I. IEEE Multimedia, **5** (1998) 78–82
2. Hofmann, M., Sabnani, K.: Streaming and broadcasting over the Internet. Proc. of the IEEE Conf. on High Performance Switching and routing, (2000) 251–256
3. Dapeng, W., Hou, Y., etc: Streaming video over the Internet: approaches and directions. IEEE Trans. on Circuits and Systems for Video Technology, **11** (2001) 282-300
4. Lu, G.: Issues and technologies for supporting multimedia communications over the Internet. Computer Communications, **23** (2000) 1323–1335
5. Abdelzaher, T., Shin, K.: QoS Provisioning with qContracts in Web and Multimedia Servers. Proc. of IEEE Real-Time Systems Symposium, (1999)
6. Reumann, J., Shin, K.: Adaptive Quality-of-Service Session Management for Multimedia Servers. Proc. of 8th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, (1998)
7. Shin, K., Abdelzaher, T., etc.: Multimedia-friendly server and communication design. IEEE Multimedia, **6** (1999) 84–90
8. Aref, W., Braun, D., etc.: An inexpensive, scalable, and open-architecture media server. Proc. First IEEE/Popov Workshop on Internet Technologies and Services, (1999) 36-43

9. Ghandeharizadeh, S., Muntz, R.: Design and implementation of scalable continuous media servers. Parallel Computing, **24** (1998) 91–122
10. Ghandeharizadeh, S., Zimmermann, R., Shi, W.: Mitra: A Scalable Continuous Media Server. Multimedia Tools and Applications, **5** (1997) 79–108
11. Blonde, L., Buck, M., etc.: A virtual studio for live broadcasting: the Mona Lisa project. IEEE Multimedia, **3** (1996) 18–29
12. ActiveX Controls, http://www.microsoft.com/com/tech/activex.asp
13. High-Availability Linux Project, http://linux-ha.org
14. Linux Virtual Server Project, http://linux-vs.org
15. Cidon, I., Rom, R., etc.: Hybrid TCP-UDP transport for Web traffic. IEEE Int. Performance, Computing and Communications Conf., (1999) 177–184
16. Stevens, W.: TCP/IP illustrated, volume 1. Addison-Wesley, New York, (1994)
17. PAXNET, http://www.paxnet.co.kr
18. LYCOS KOREA, http://www.lycos.co.kr

On the Development of an Information Service System for Navigational Safety of Seagoing Vessels

Gilyoung Kong[1], Chungro Lee[2], Soonkap Kim[1],
Gilsoo Kim[3], and Cholseong Kim[4]

[1] Division of ship operation systems engineering,
Korea Maritime University, Pusan, Korea
{gykong,soonkap}@kmaritime.ac.kr
[2] Division of education and training,
Korea Institute of Maritime and Fisheries Technology, Pusan, Korea
[3] Division of maritime transportation science,
Korea Maritime University, Pusan, Korea
[4] Institute of Maritime Industry,
Korea Maritime University, Pusan, Korea

Abstract. This study aims to develop the information service system of navigational safety for seagoing vessels. It has been successful in developing the intended system which has made possible to provide the following functions and capabilities:

- immediate computation of the dynamic motions of a ship against the present and future weather conditions;
- evaluation of the integrated seakeeping performance of a ship on a real time basis and providing the navigators with the navigational information on the spot enabling them to take the most appropriate countermeasures; and
- supporting the navigators in the selection of a safe sailing route by providing the integrated seakeeping performance.

1 Introduction

Ship operators navigating in rough seas should take grip with the extent of navigational safety and take measures according to the condition of ship itself, weather and sea in order to secure safe and economic operation of the ship.

This study aims at developing navigation safety information system through internet for the purpose of preventing sea casualties in rough seas and economic operation of the ship. First of all, a ship's operator should key in the following information:

- current weather information of the area in which the ship is navigating,
- the future information of the area in which the ship shall navigate,
- the ship's principal dimension and the condition of the body of the ship through internet network.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 273–283, 2002.

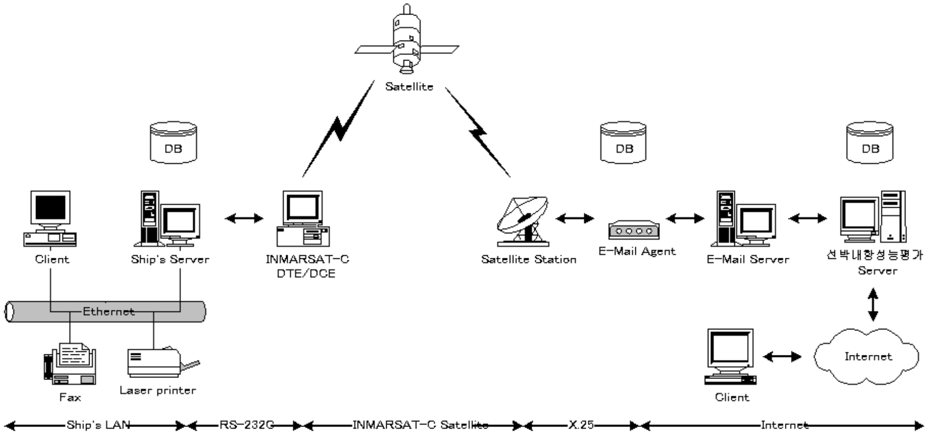


Fig. 1. Schematic diagram of the system for internet service

Based on this information, navigation safety information system evaluates, in real time, the degree of navigation safety by calculating the response of ship's motion in the current position and the future estimated position. The evaluated information is provided to the ship through internet network to take some appropriate measures by the operators, and furthermore another evaluation on the basis of weather forecasting will be done and allow the operators to choose safer route.

2 The Structure of Navigation Information System

2.1 Internet Network for Providing Information

Some information about the current weather information of the area in which the ship is navigating, the future information of the area in which the ship shall navigate, the ship's principal dimension and the condition of the body of the ship should be typed in, in real time, through internet network. The information is processed by the system, which is not available on board the ship, and is fed back into the ship through internet network which also takes place in real time. The overall configuration of the system is depicted in Fig. 1.

The flow of the data processing system which gets real-time data from a ship and evaluates the degree of the navigation safety is illustrated in Fig. 2.

The picture frame of input and output is given as Fig. 3. The input information is the current weather information of the area in which the ship is navigating, the future information of the area in which the ship shall navigate, the ship's principal dimension and the condition of the body of the ship through internet network. The output informations available by the operator of the ship are the degree of risk against the navigation safety of the ship and the index of navigational safety.

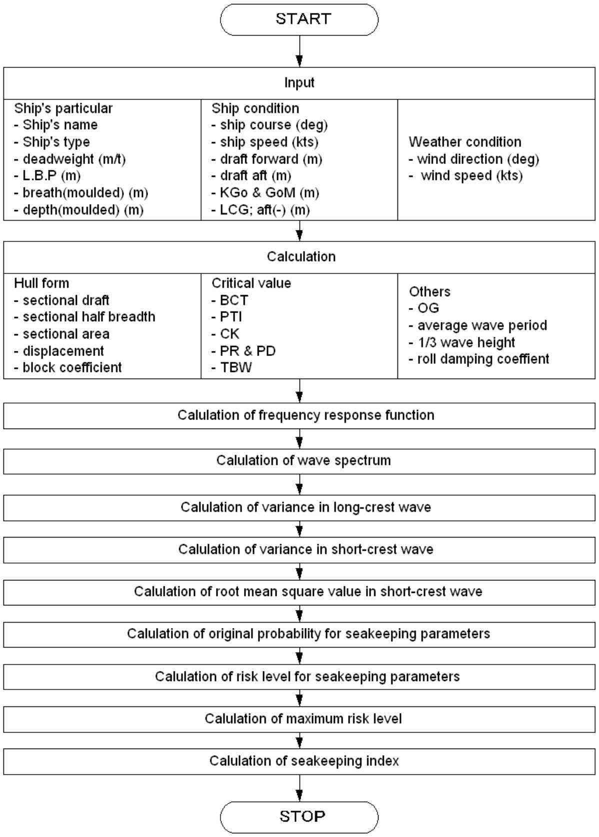


Fig. 2. Flow chart of navigational safety evaluation program

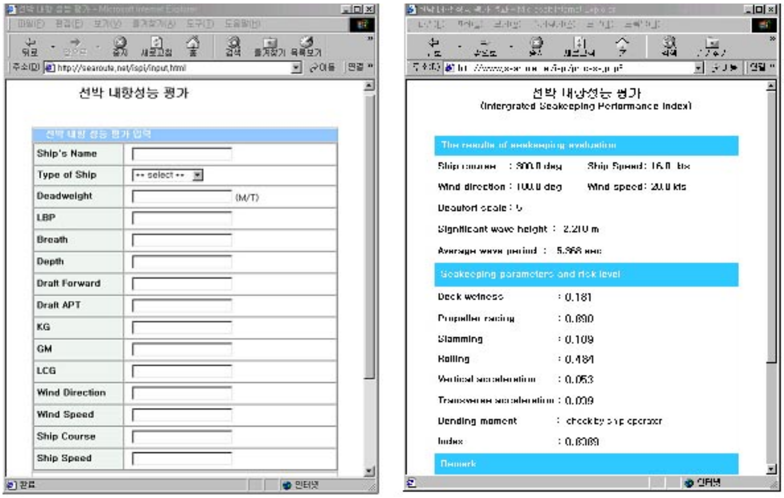
2.2 The Communication between Ship and the System

When the data given by the ship's operator is sent to the server of navigation safety information system, the server performs evaluation and the result is fed back to the ship. For the purpose of file transfer between ships and shore, INMARSAT-C facilities play roles as DTE/DCE(data terminal equipment/data circuit terminating equipment). Both earth station of the satellite and INMARSAT-C satellite connect those between.

3 The Evaluation of Real-Time Navigation Safety

3.1 Evaluated Factors of Navigation Safety

In order to evaluate the degree of navigational safety of a ship with special consideration to the people on board, ship's body and cargo of the ship, that is navigating rough and irregular sea condition, the following factors are considered for the evaluation [1].



선박 내항성능 평가

선박 내항 성능 평가 입력

Ship's Name	
Type of Ship	-- select --
Deadweight	(M/T)
LBP	
Breath	
Depth	
Draft Forward	
Draft APT	
KG	
GM	
LCG	
Wind Direction	
Wind Speed	
Ship Course	
Ship Speed	

선박 내항성능 평가
(Intergrated Seakeeping Performance Index)

The results of seakeeping evaluation

Ship course : 300.0 deg Ship Speed : 16.0 kts

Wind direction : 100.0 deg Wind speed : 20.0 kts

Beaufort scale : 5

Significant wave height : 2.210 m

Average wave period : 5.968 sec

Seakeeping parameters and risk level

Deck wetness : 0.181

Propeller racing : 0.090

Slamming : 0.109

Rolling : 0.484

Vertical acceleration : 0.053

Transverse acceleration : 0.009

Bending moment : check by ship operator

Index : 0.6069

Remark

Fig. 3. Data input (*left*) and output (*right*) format on the web

(1) deck wetness
(2) propeller racing
(3) slamming
(4) rolling
(5) vertical acceleration
(6) transverse acceleration
(7) bending moment

The systemic combination of those evaluated factors has the form of serial combination. If the probability of occurrence of just one factor exceeds the critical occurrence probability, then the overall seakeeping performance fails and the ship may be endangered [2].

3.2 Variance of Navigation Safety Evaluated Factors

When a ship is navigating on sea which has single wave distance and irregularity and is keeping constant course(χ) and speed(V), spectrum $S_{x_i}(\omega_e,\chi)$ is as follows if we put X_i(t) (the random process of evaluation factors calculated from the response function of ship's motion) as $H_{x_i}(\omega_e$,V,χ-θ):

$$S_{x_i}(\omega_e, \chi) = \int_{-\pi/2}^{\pi/2} \left| H_{x_i}(\omega_e, V, \chi - \theta) \right|^2 S_\zeta(\omega_e, \chi) d\theta \tag{1}$$

Variance $\sigma^2_{x_i}$ is as the following.

$$\sigma^2_{x_i}(\chi, V, S) = \int_0^\infty S_{x_i}(\omega_e, \chi) d\omega_e \tag{2}$$

The variables that let the variance in formula (2) change are the encountering angle of the ship and the wave, the ship's speed(V), and the condition of the sea(S). And also X_i(t) is calculated as [3]:

$$X_i(t) = \int_0^\infty cos(\omega_e t + \psi_i)\sqrt{2S_{x_i}(\omega_e, \chi)d\omega_e} \tag{3}$$

Where, $\psi_i = \epsilon_i(\omega)+\gamma_i$

$\gamma_i =$ phase angle distributed uniformly between 0 and 2π

3.3 The Critical Occurrence Probability of the Evaluated Factors

The change of the modulation width against an instant time of the X_i(t) has the form of Gauss distribution and the extreme value has the form of Rayleigh distribution [4]. Once the variance $\sigma^2_{x_i}$ is acquired, then (which means the probability that the extreme value of X_i(t) exceeds a constant value X_i) is as follows:

$$Q_{X_i} = \int_{X_1}^\infty (\frac{X_i}{\sigma_{X_i}})exp(-\frac{{X_i}^2}{2\sigma^2_{X_i}})dX = exp(-\frac{{X_1}^2}{2\sigma^2_{X_i}}) \tag{4}$$

And then σ_{x_i} can be expressed as:

$$\sigma_{X_i} = \sqrt{-\frac{{X_1}^2}{2lnQ_{X_i}}} \tag{5}$$

When we consider critical probability $Q_{X_{ic}}$ (i.e. the probability of exceeding X_{ic}), then variance $\sigma_{x_{ic}}$ that is a value of danger can be found:

$$\sigma_{X_{ic}} = \sqrt{-\frac{{X_{ic}}^2}{2lnQ_{X_{ic}}}} \tag{6}$$

3.4 Evaluation Value of the Evaluated Factors

The extreme value of the evaluated factors shows the form of Rayleigh distribution and the occurrence probability is expressed as $Q(\overline{X_i})$. In this case E_{X_i} is defined as the evaluation value of X_i factors and is expressed as an inverse number with no dimension [2].

$$E_{X_i} = \frac{1}{\sqrt{-2ln\{Q(\overline{X_i})\}}} = \frac{\sigma_{X_i}}{\overline{X_i}} \tag{7}$$

When the evaluation value E_{X_i} becomes zero, then the confidence level of the random factor (X_i) is 1.0 and when the evaluation value E_{X_i} becomes infinite then the confidence level of the random factor (X_i) is zero.

3.5 The Risk of the Evaluated Factors

We define $E_{X_{ic}}$ as the critical evaluation value done on critical occurrence probability of random factor of X_i, and put μ_{X_i} as the following:

$$\mu_{X_i} = \frac{E_{X_i}}{E_{X_{ic}}} = \frac{[\overline{X_i}/\sigma_{X_{ic}}]}{[\overline{X_i}/\sigma_{X_i}]} = \frac{\sigma_{X_i}}{\sigma_{X_{ic}}} \tag{8}$$

On the other hand, when $\mu_{X_i} \geq 1.0$, then X_i(the factor of seakeeping performance) becomes dangerous, and when $\mu_{X_i} < 1.0$, it shows that the ship is safe.

3.6 Development of Evaluation Index of Navigational Safety

After investigating which critical occurrence probability of a factor is affecting the most to the overall critical occurrence probability, if the factor has the same degree of danger ('propeller racing' is found to be the largest, i.e. Q_{pc}=0.1), transformed evaluation value ($\tilde{E_i}$) is found from the evaluation value (E_i) of each factor [5].

(1) In case of 'propeller racing'

$$\tilde{E_p} = \frac{E_p}{\alpha_{pp}} = E_p \cdot \frac{E_{pc}}{E_{pc}} = E_p \tag{9}$$

Where, E_p: Evaluation value of propeller racing $\left(\frac{\sigma_p}{X_p^*} = \frac{1}{\sqrt{-2ln(Q_p)}}\right)$

$\tilde{E_p}$: transformed evaluation value of propeller racing

E_{pc}: Critical evaluation value of propeller racing $\left(\frac{\sigma_{pc}}{X_p^*} = \frac{1}{\sqrt{-2ln(0.1)}}\right)$

(2) Other than 'propeller racing'

$$\tilde{E_i} = \frac{E_i}{\alpha_{pi}} = E_i \cdot \frac{E_{pc}}{E_{ic}} = E_{pc} \cdot \mu_i, \tilde{E_j} = \frac{E_j}{\alpha_{pj}} = E_j \cdot \frac{E_{pc}}{E_{jc}} = E_{pc} \cdot \mu_j \tag{10}$$

where, α_{pi}: ratio of critical evaluation value between propeller racing and i factor, μ_i: dangerousness of i factor

On the other hand, when the degree of danger is the same, then transformed values remain the same, and the occurrence probability(Q_i) shall have the same value.

$$\mu_i = \mu_j \rightarrow \tilde{E_i} = \tilde{E_j}, \tilde{Q_i} = \tilde{Q_j} \tag{11}$$

If we assume that each element of the factors be independent for the purpose of finding the overall occurrence probability, then the following evaluation index is identified.

$$\tilde{E_T} = \frac{1}{\sqrt{-2ln(1 - \tilde{P_T})}} \tag{12}$$

Where, $\tilde{P_T} = \prod_{i=1}^{n} \tilde{P_i}$

$$\begin{aligned}\tilde{P_i} &= 1 - exp\left\{-\tfrac{1}{2}\left(\tfrac{1}{\tilde{E_i}}\right)^2\right\} \\ &= 1 - exp\left\{-\tfrac{1}{2}\left(\tfrac{\alpha_{pi}}{E_i}\right)^2\right\} \\ &= 1 - exp\left\{-\tfrac{1}{2}\left(\tfrac{\alpha_{pi} \cdot {X_i}^*}{\sigma_i}\right)^2\right\} \\ &= 1 - Q({X_i}^*)^{{\alpha_{pi}}^2}\end{aligned} \qquad (13)$$

$$E_{Tc} = \frac{1}{\sqrt{-2ln(1 - P_{Tc})}} \qquad (14)$$

Where, $P_{Tc} = \prod_{i=1}^{n} P_{ic}$

$P_{ic} = 1 - exp\left\{-\tfrac{1}{2}\left(\tfrac{{X_i}^*}{\sigma_{ic}}\right)^2\right\} = 1 - Q_{ic}$

P_{Tc}: Confidence level function of seakeeping performance
$Q(X_i)$: Occurrence probability of each factor
Q_{ic}: Critical occurrence probability of each factor (Rayleigh distribution)

In order to find out the overall degree of danger of the system, the ratio of formula (14) to formula (12) can be expressed as $\tilde{\mu_T}$.

$$\tilde{\mu_T} = \frac{\tilde{E_T}}{E_{Tc}} = \sqrt{\frac{ln(1 - P_{Tc})}{ln(1 - \tilde{P_T})}} \qquad (15)$$

If $\tilde{\mu_T}$ is greater than 1.0, the total system is judged to be in danger. When just one factor of $\tilde{\mu_T}$ becomes more than 1.0, $\tilde{\mu_T}$ tends to be greater than 1.0.

4 Discussion on Navigation Safety Evaluated from the Model

Fig. 4 ∼ 6 shows the calculation result that when 2,641 TEU container ship navigates with the speed of 12 knots, 16 knots, 20 knots respectively against wind of 40 knots, with the interval of 15° starting from the head-on wave.

If the ship navigates with the speed of 12 knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steering course remains inside the scope of 0° (head-on) to 60°. And it is evaluated that the ship is in danger, particularly due to deck-wetness and vertical acceleration.

When the ship navigates with higher speed of 16knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steering course remains inside the scope of 0° (head-on) to 70°. And it is evaluated that the ship is in danger more widely, particularly due to deck-wetness and vertical acceleration.

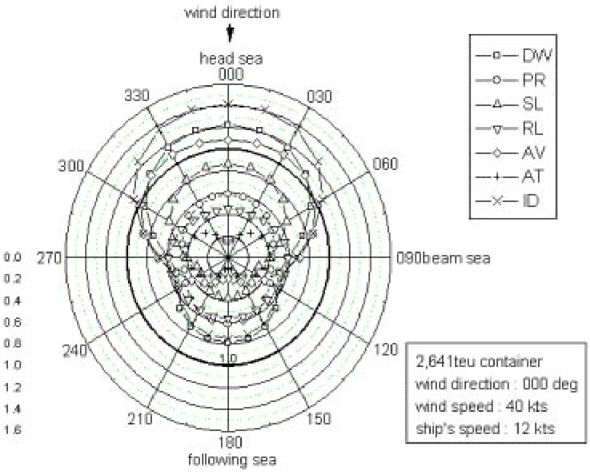


Fig. 4. Seakeeping evaluation of 2,641TEU container(speed=12 knots)

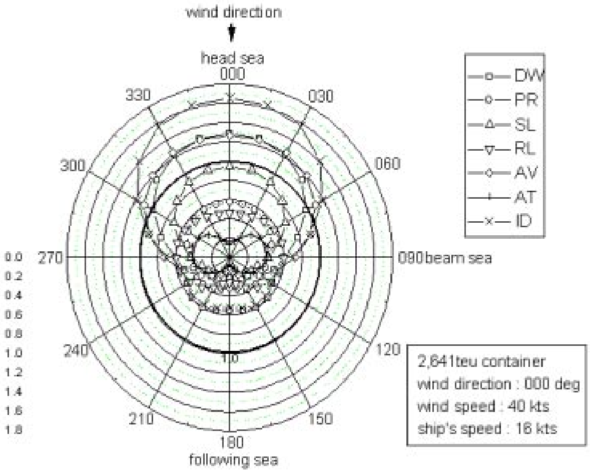


Fig. 5. Seakeeping evaluation of 2,641TEU container(speed=16 knots)

When the ship navigates with speed of 20 knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steering course remains inside the scope of 0° (head-on) to 75°. And it is evaluated that the ship is in danger much more widely, due to deck-wetness and vertical acceleration as well as slamming.

Fig. 7 ∼ 9 shows the calculation result that when 4,000 TEU container ship navigates with the speed of 12 knots, 16 knots, 20 knots respectively against wind of 40 knots, with the interval of 15° starting from the head-on wave.

When the ship navigates with the speed of 12 knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steer-

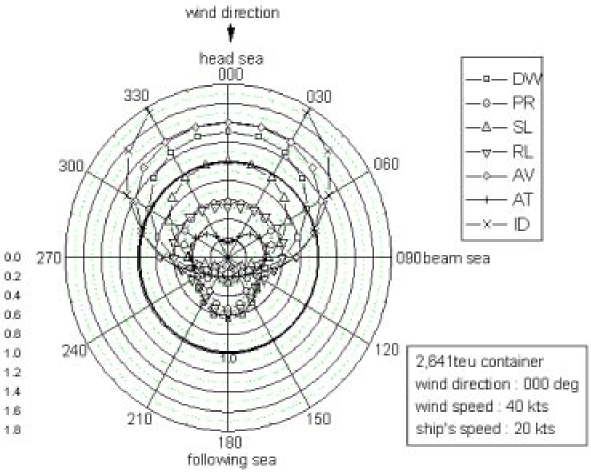


Fig. 6. Seakeeping evaluation of 2,641TEU container(speed=20 knots)

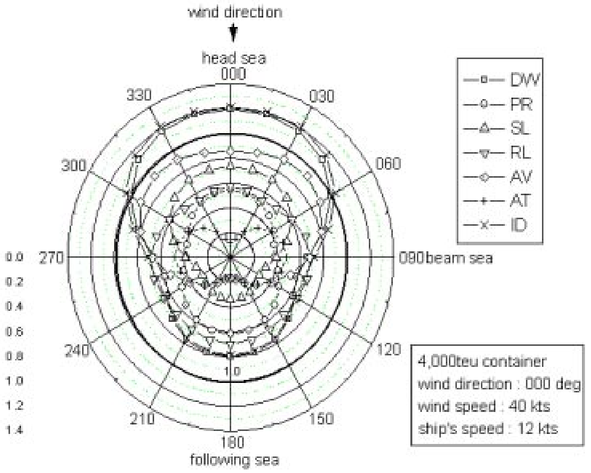


Fig. 7. Seakeeping evaluation of 4,000TEU container(speed=12 knots)

ing course remains inside the scope of 0° (head-on) to 60°. And it is evaluated that the ship is in danger, particularly due to deck wetness only.

When the ship navigates with higher speed of 16knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steering course remains inside the scope of 0° (head-on) to 70°. And it is evaluated that the ship is in danger more widely, particularly due to deck-wetness and vertical acceleration.

When the ship navigates with speed of 20 knots, the index of seakeeping performance exceeds 1 when the encountering angle of wave and the ship's steering course remains inside the scope of 0° (head-on) to 75°. And it is evaluated that

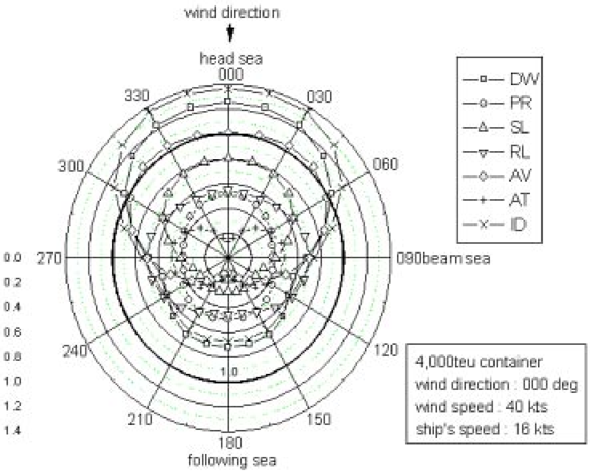


Fig. 8. Seakeeping evaluation of 4,000TEU container(speed=16 knots)

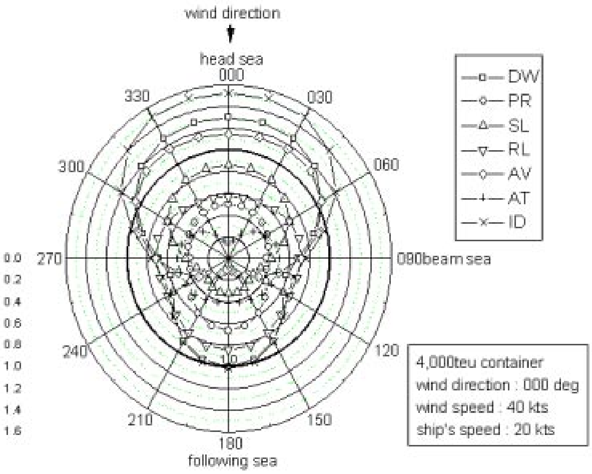


Fig. 9. Seakeeping evaluation of 4,000TEU container(speed=20 knots)

the ship is in danger much more widely, due to deck-wetness and vertical acceleration as well as slamming.

The overall evaluation on a container ship reveals that the index of seakeeping performance is the highest in case of head-on wave, and it reduces gradually as the wave direction changes to beam direction. As a result, the index has the lowest figure in case of beam wave. And it also shows that the speedier the ship the higher the index is.

This experiment finally advises that speed reducing and changing course is one of desirable way of action when the ship meets rough seas.

5 Conclusion

This study has developed a system that can evaluate navigational safety of a ship in real time on the basis of the following information:

- the current weather information of the area in which the ship is navigating,
- the future information of the area in which the ship shall navigate,
- the ship's principal dimension, and
- the condition of the body of the ship through internet network.

The followings are suggested and may be implemented.

1) A ship operator can get information about the evaluated information on navigational safety in real time by being connected to the system through internet.
2) In addition to the operator's own experience, safer navigation can be achieved by using the information generated from this system.
3) Ship management division in shore may make use of this system for the management of the ships concerned, fleet management and ship operation contract.
4) Optimal route forecasting may come true by using this system on top of ocean route service.

References

1. Kishev, R.: General Considerations on Ship Seakeeping Optimization in Design. Osaka Meeting on Seakeeping Performance, Japan (1992) 217–242
2. KIM, S.K., Naito, S., Nakamura, S.: The Evaluation of Seakeeping Performance of a Ship in Waves. Journal of the Society of Naval Architects of Japan, **155** (1984) 71–83
3. Pierson, W.J. Jr., Neumann, G., James, R.W.: Practical Methods for Observing Ocean Waves by Means of Wave Spectra and Statistics. U.S. Navy Hydrographic Office Pub., No.603 (1955)
4. Rice, S.O.: Mathematical Analysis of Random Noise. The Bell System Technical Journal, **24** (1945)
5. KIM, S.K., Kong, G.Y.: A Study of the Integrated Seakeeping Performance Index in Seaway. Journal of The Korean Institute of Navigation, **21** (1997) 1–10

Optimal Code Weight Selection for 2-D Optical CDMA Systems*

Taewon Kim[1], Wonjin Sung[1], and Jinwoo Choe[1]

Department of Electronic Engineering, Sogang University,
1 Sinsu-dong, Mapo-gu, Seoul 121-742, Korea
{dragonix,wsung,xinu}@sogang.ac.kr

Abstract. This paper presents the detection performance of the two-dimensional optical CDMA network under varying conditions of the bit-level design parameters such as the chip length (M), the number of wavelengths in use (L), and the code weight (W). In particular, the code weight that results in the minimal BER is analytically determined as $W \approx 1.4LM/m$ under the assumption of sparsely distributed random code selection, where m represents the number of active nodes in the network. The derived expression is shown to closely match the optimal code weight obtained via simulation for the cases of practical interest. The BER performance using the optimal code weight when the number of active nodes is not known (and hence estimated) is also presented and compared with the optimal performance.

1 Introduction

The era of optical local area networking is expected to begin in the near future, as fascinating (and bandwidth-exhaustive) network applications, such as multimedia, 3-D graphics, and virtual reality applications, call for bandwidths that electronic devices can hardly support at a reasonable cost. In contrast to the *Wide Area Networks* (WANs) or nationwide backbone networks, *Local Area Networks* (LANs) may be characterized by frequent changes of configuration and bursty transmission behavior of network nodes. Consequently, *Wavelength Division Multiplexing* (WDM) and *Time Division Multiplexing* (TDM) which require sophisticated network dimensioning and resource allocation, may not be adopted to LANs as successfully as to WANs. The *Optical Code Division Multiple Access* (OCDMA) LAN architecture proposed in [1,2,3] is considered to be a more favorable solution, and a significant amount of research effort has been devoted to code design and detection problems in OCDMA LANs [4,5,6,7,8].

For the lack of practical means for optical phase modulation/demodulation, *Intensity Modulation and Direct Detection* (IM/DD) is assumed in most studies on OCDMA LANs (see [9, Table I] for a concise comparison between coherent OCDMA and incoherent OCDMA). This implies that OCDMA codes are

* This work was supported by the Korea Science Engineering Foundation under Grant No. R01-2001-00542.

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 285–294, 2002.

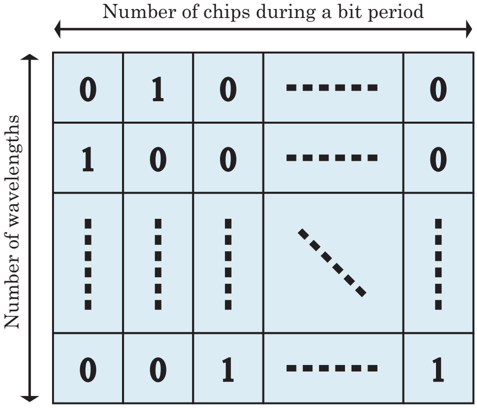


Fig. 1. Matrix representation of a 2-D OCDMA code

usually generated by locating a certain number of optical pulses (possibly with different intensities) at different positions along time axis and/or wavelength axis [10,2,11]. The class of OCDMA techniques based on 2-D (wavelength and time) codes are often referred to collectively as 2-D OCDMA [12,13,14]. In general, 2-D codes for OCDMA can be represented in a matrix form as illustrated in Figure 1, where the entries of the matrix corresponds to the pulse intensity located at different chip periods in different colors (i.e., wavelengths). Due to the 2-D structure and the unipolarity of 2-D OCDMA codes, traditional code design and detection techniques for wireless CDMA systems may not be directly applicable for the 2-D OCDMA code design and detection [12]. As a result, several new code sets for 2-D OCDMA and their optimal detection methods have been proposed [7,15,16]. In particular, a *Maximum-Likelihood* (ML) detector is proposed based on a simple *Multi-Access Interference* (MAI) model which is applicable to a wide range of OCDMA LANs, and its BER performance is investigated analytically and experimentally in [15,17].

In this paper, we extend the study in [15,17] under the identical set of fundamental assumptions, and investigate the relationship between the code weight (i.e., the total number or intensity of optical pulses in individual 2-D OCDMA codes) and the BER performance of the ML detector. In particular, the optimal code weight for the minimized BER is derived from the known BER expression and experimentally verified using computer simulation. The optimal code weight is expressed as a simple function of m, L, and M representing the number of active nodes (i.e., nodes transmitting data frames), the number of wavelengths used for 2-D OCDMA, and the number of chips during a bit period, respectively. It should here be noted that L and M are usually considered as a system parameter of a 2-D OCDMA LAN, which is known to all nodes beforehand. Therefore, by accurately estimating m which generally changes with time, individual active nodes can adjust their code weight dynamically in cooperation to reduce their BER. Also, the ML detector proposed in [15,17] contains m in its decision rule, and hence, the anticipated BER performance of the ML detector will also be achievable only if an accurate estimate of m is provided. For these reasons, accurately estimating the number of active nodes in a OCDMA LAN must be considered as a critical task to improve the BER performance of the LAN. In this paper, we also address the problem of estimating m based only on the optical signal observed at a node over a time period without any explicit exchange of information between nodes on their activities. In particular, the minimum length of observation period to guarantee desired accuracy of the estimate will be investigated through numerical experiments.

2 Interference Model and ML Detector for OCDMA LANs

In this section, we briefly review the set of fundamental assumptions made in [15] for the study of 2-D OCDMA LAN performance, including the MAI model. In [15], all nodes in a 2-D OCDMA LAN are assumed to be interconnected through one or more ideal star couples. This means that every node will be exposed to optical signals transmitted by all the other nodes, and hence, the optical signal received by a node over each wavelength channel can be expressed as a (possibly delayed) superposition of all optical pulses transmitted by nodes in the LAN over the wavelength channel. As was done in [15], we will assume that each pair of communicating nodes are under perfect bit synchronization, and all interfering pulses are chip-synchronized with the signal being interfered. As a result, the optical signal received by a node over a bit period can be expressed by a matrix R with non-negative integer entries $r_{i,j}$ ($i = 1, \ldots, L$ and $j = 1, \ldots, M$), which is either the superposition of interfering optical pulses or that of a 2-D OCDMA code and interfering pulses depending on the bit transmitted by the transmitting node; i.e.,

$$R = \begin{cases} X & \text{if “0” is transmitted,} \\ C + X & \text{if “1” is transmitted,} \end{cases} \tag{1}$$

where C represents the code that the receiving node is supposed to detect when “1” is transmitted by the transmitting node, and X the MAI from other nodes. Consequently, the optimal detection method depends on the stochastic model for the interfering signal X, and we next introduce the interference model used in [15].

In [15], it is assumed that an interfering node transmits a pulse at a position (i, j) with probability of $W/2LM$, where W is the weight of 2-D OCDMA codes[1]. Therefore, assuming independence between interfering nodes, the distribution of pulse intensity of the interfering signal X at position (i, j) is given by the following binomial distribution:

$$P[X_{i,j} = x_{i,j}] = \binom{m-1}{x_{i,j}} \left(\frac{W}{2LM}\right)^{x_{i,j}} \left(1 - \frac{W}{2LM}\right)^{m-x_{i,j}-1} . \tag{2}$$

[1] In other words, all 2-D OCDMA codes used in the LAN are composed of exactly W optical pulses located at different positions in the $L \times M$ matrix.

In [15], the independence between $X_{i,j}$'s is further assumed for the derivation of their joint distribution. In other words, the joint distribution of the interference matrix X is approximated simply by the product of (2),

$$P[X = x] = \prod_{\substack{i=1,\dots,L \\ j=1,\dots,M}} \binom{m-1}{x_{i,j}} \left(\frac{W}{2LM}\right)^{x_{i,j}} \left(1 - \frac{W}{2LM}\right)^{m - x_{i,j} - 1} . \tag{3}$$

From (1) and the interference model (i.e., the joint distribution of X), one can easily derive the distribution of R under the condition that "0" (or "1") is sent. By comparing these conditional distributions of R, the decision rule of the ML detector can be obtained as follows:

$$\prod_{(i,j)\in E_1} \left(\frac{m - R_{i,j}}{R_{i,j}}\right) \underset{\text{"0 detected"}\atop >}{\overset{\text{"1 detected"}}{<}} \left(\frac{2LM}{W} - 1\right)^{W} , \tag{4}$$

where $E_1 = \{(i,j) : C_{i,j} = 1\}$. This ML detector is referred to as the "AND" detector in [15] because the decision rule can be closely approximated by the logical "AND" operation on the existence of pulses at positions E_1. More precisely, when $2mW$ is not much larger than LM, the ML decision rule is virtually identical to the following decision rule:

$$\text{if } \wedge_{(i,j)\in E_1} Y_{i,j}, \text{ then "1" is detected, and "0" is detected otherwise,} \tag{5}$$

where $Y_{i,j}$ is the event that one or more pulses are detected at position (i,j). As a result, the BER performance of the ML detector should not be very different from that of the genuine AND detector given by (5). The BER of the genuine AND detector can be easily obtained from the probability that one or more interfering pulses are located at every $(i,j) \in E_1$ as

$$\begin{aligned} \text{BER} &= \frac{1}{2} \prod_{(i,j)\in E_1} \sum_{k=1}^{m-1} \binom{m-1}{k} \left(\frac{W}{2LM}\right)^{k} \left(1 - \frac{W}{2LM}\right)^{m-k-1} \\ &= \frac{1}{2} \left(1 - \left(1 - \frac{W}{2LM}\right)^{m-1}\right)^{W} . \end{aligned} \tag{6}$$

Note that if $W \ll LM$ and $m \gg 1$, then the BER of the ML detector (as well as that of the genuine AND detector) can be further approximated by

$$\text{BER} \approx \frac{1}{2} \left(1 - e^{-\frac{Wm}{2LM}}\right)^{W} . \tag{7}$$

3 Optimal Code Weight

We observe that the expression in (7) takes the form of

$$\text{BER} \approx \frac{1}{2} \left\{ \left(1 - e^{-x}\right)^{x} \right\}^{\frac{2LM}{m}} = \frac{1}{2}\, y^{\frac{2LM}{m}} \tag{8}$$

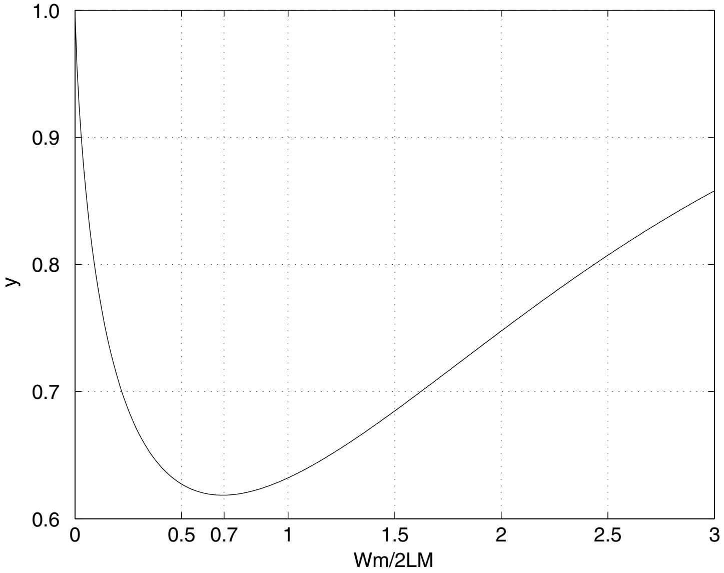


Fig. 2. Evaluation of the function $y = \left(1 - e^{-x}\right)^x$, where $y \propto$ BER and $x = Wm/(2LM)$

where $x = Wm/(2LM)$ and $y = (1 - e^{-x})^x$. The BER expression in (8) is minimized for varying values of W when y is minimized. Evaluation of y as a function of x is shown in Fig. 2, which indicates the minimum of y occurs at around $x = 0.7$. The exact value of x that minimizes y can be determined by taking the derivative of y as

$$\frac{dy}{dx} = \left(1 - e^{-x}\right)^x \left(\ln(1 - e^{-x}) + \frac{xe^{-x}}{1 - e^{-x}}\right) \tag{9}$$

and $dy/dx = 0$ is obtained by $x = \ln 2 = 0.69$. Thus the code weight that minimizes the BER for given values of L, M and m is determined from $x = Wm/(2LM) = 0.69$ as

$$W_{opt} = 1.39 \times \frac{LM}{m} \simeq 1.4 \times \frac{LM}{m}. \tag{10}$$

The factor $K = mW_{opt}/(LM) \simeq 1.4$ has the interpretation of the "average light intensity" per unit square in Fig. 1, which is a chip duration for a given wavelength color. I.e., the code weight should be chosen such that 1.4 optical pulses (or light intensity) are present on the average per unit square in the matrix representation of the 2-D code.

The shape of the curve plotted in Fig. 2 determines a general behavior of the BER performance as the code weight changes. In Fig. 3, the BER expression in (6) is evaluated and plotted for varying code weights. The dimension of the code, defined as $N = LM$ (the total number of unit squares in the matrix

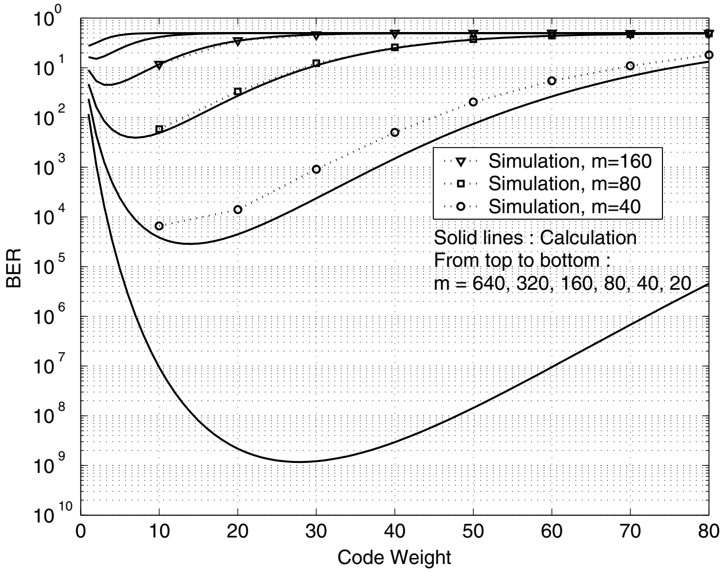


Fig. 3. Bit error rate performance for varying values of the code weight ($N = 400$)

representation of the code), is 400 for the figure. Different numbers of active nodes have been used for the BER evaluation. As anticipated from eq. (10), the code weight that produces the minimum BER increases as the number of active nodes decreases. Simulation results for the BER performance are plotted together in the figure for comparison. The simulation employed a random selection of 2-D codes for m active nodes and performed detection using the test described in (4). Similar BER calculation and simulation results are shown in Fig. 4 for the case of code dimension of $N = 900$. For the same number of active nodes, a decreased BER performance results when the code dimension increases.

Detection performance has been evaluated for various different values of the code dimension and the number of active nodes, and the optimal code weight for each of the cases considered has been experimentally determined. Fig. 5 shows the optimal code weight determined for varying values of N and m. The average light intensity per unit square can then be obtained for the optimal code weight using the relation $K = mW_{opt}/(LM)$. Table 1 summarizes the experimentally obtained values of W_{opt} and K. As can be seen from the table, the optimal weight and the average light intensity are in good agreement with the analytically determined result given in eq. (10), for a wide range of design parameters of practical interest.

4 Estimation of the Number of Active Nodes

Although the determination of the optimal code weight requires the knowledge of the number of active nodes (or active users), such information may not be

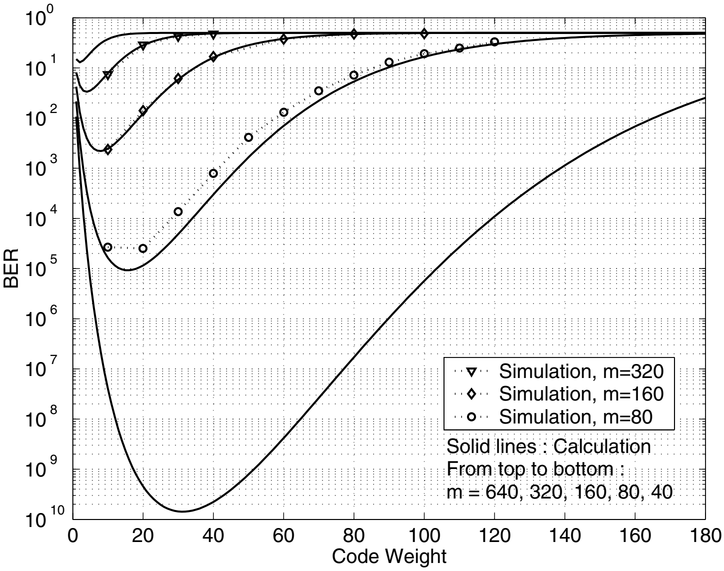


Fig. 4. Bit error rate performance for varying values of the code weight ($N = 900$)

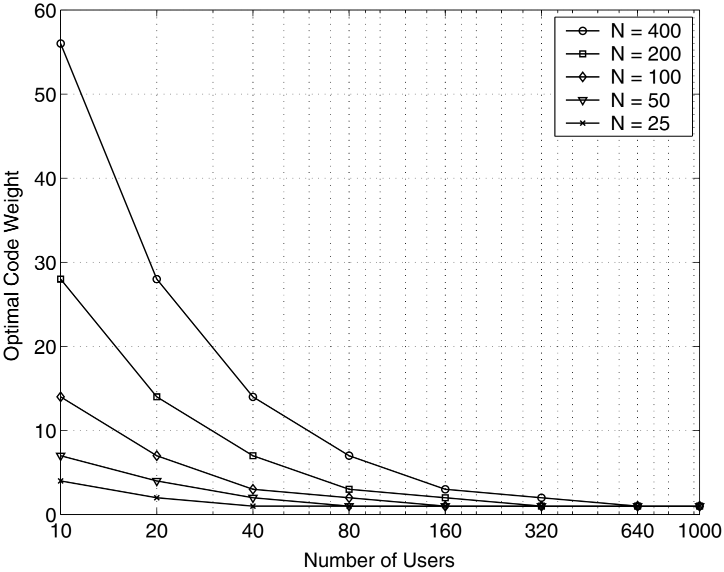


Fig. 5. Optimal code weight versus the number of active nodes

available to each node. Furthermore, the number of active nodes is often a time-varying quantity and need to be estimated by each node for dynamic adjustment of its code weight for improved detection performance.

Table 1. Optimal code weight and average light intensity

N	m	W_{opt}	K
50	10	7	1.400
100	20	7	1.400
200	40	7	1.400
400	80	7	1.400
900	160	8	1.422
100	10	14	1.400
200	20	14	1.400
400	40	14	1.400
900	80	16	1.422
10000	500	28	1.399

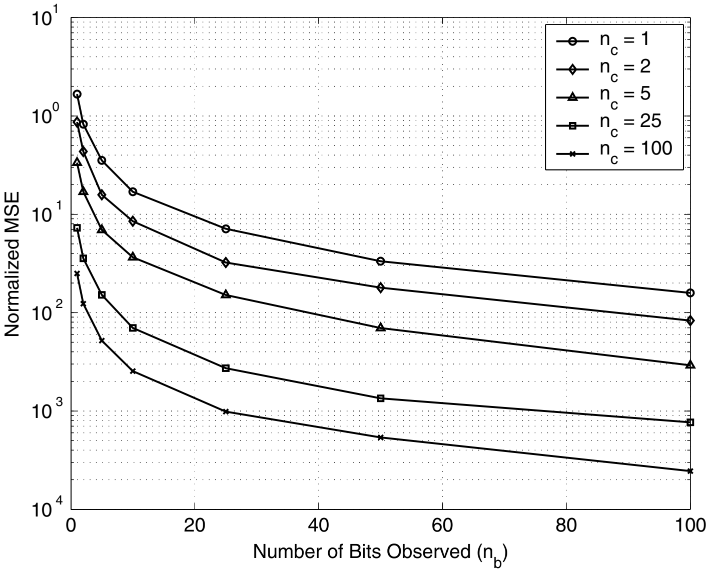


Fig. 6. Estimation performance for the number of active nodes ($N = 100$, $m = 20$)

We consider a very simple procedure for the estimation and evaluate the BER based on such estimation of the number of active nodes. Let n_c and n_b respectively represent the number of unit squares per bit and the number of bits that are observed by a node. The node simply adds up the light intensity that it observes in n_c unit squares for the duration of n_b bits. Then it divides the added intensity by $n_c n_b W$, where W is the code weight used in the system at the time of observation. As n_c and n_b increases, the estimation will become more accurate. Fig. 6 shows the normalized mean square error (MSE), which is a fast decreasing function of n_c and n_b. The result has been obtained using the code dimension $N = 100$ and the true number of active nodes $m = 20$. The BER

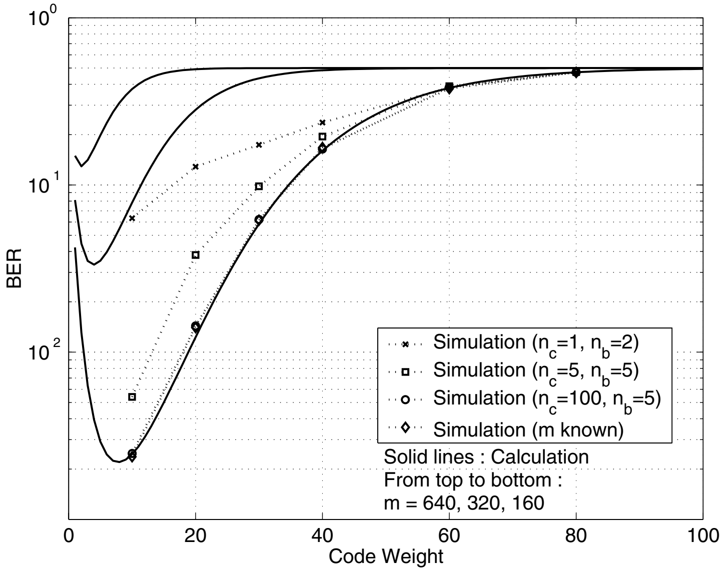


Fig. 7. Bit error rate performance for varying values of the code weight using the estimate for the number of active nodes ($N = 900$)

performance based on the node number estimation is simulated and shown in Fig. 7. The BER for the case of known m is identical to the result shown in Fig. 4. By increasing n_c and n_b values, the performance based on estimation (indicated by dotted lines in the figure) quickly approaches the desired BER level using a known number of active nodes (indicated by solid lines in the figure).

5 Conclusion

The optimal code weight that minimizes the detection error probability of the 2-D OCDMA networks has been derived, and the resulting BER performance has been presented for various cases of bit-level design parameters. The average light intensity per unit square in the matrix representation of the code is shown to be 1.4, which is both determined from the derived expression of the optimal weight and confirmed by simulation results. The BER performance using a simple estimation scheme for an unknown number of active nodes has been also presented.

References

1. Salehi, J.A.: Code division multiple access techniques in optical fiber networks – Part I: Fundamental principles. IEEE Transactions on Communications **37** (1989) 824–833

2. Salehi, J.A.: Emerging optical code-division multiple access communication systems. IEEE Nerwork **3** (1989) 31–39
3. Foschini, G.J., Vannucci, G.: Using spread-spectrum in a high-capacity fiber-optic local network. Journal of Lightwave Technology **6** (1988) 370–379
4. Andonovic, I., Tancevski, L., Shabeer, M., Bazgaloski, L.: Incoherent all-optical code recofnition with balanced detection. IEEE Journal of Lightwave Technology **12** (1994) 1073–1080
5. Chung, F.R.K., Salehi, J.A., Wei, V.K.: Optical orthogonal codes: Design, analysis, and applications. IEEE Transactions on Information Theory **35** (1989) 595–604
6. Nelson, L.B.k Poor, H.V.: Performance of multiuser detection for optical CDMA-Part I: Error probabilities. IEEE Transactions on Communications **43** (1995) 2803–2811
7. Shivaleela, E.S., Sivarajan, K.N., Selvarajan, A.: Design of a new family of two-dimensional codes for fiber-optic CDMA netorks. Journal of Lightwave Technology **16** (1998) 501–508
8. Yang, G.C., Kwong, W.C.: Two-dimensional spatial signature patterns. IEEE Transactions on Communications**44** (1996) 184–191
9. Kitayama, K.: Code division multiplexing lightwave networks based upon optical code conversion. IEEE Journal on Selected Areas in Communications **16** (1998) 1309–1319
10. Lee, S.W., Green, D.H.: Performance analysis method for optical codes in coherent optical CDMA networks. IEE Proceedings of Communications **147** (2000) 41–46
11. Yang, G.C, Kwong, W.C.: Performance comparison of multiwavelength CDMA and WDMA+CDMA for fiber-optic networks. IEEE Transactions on Communications **45** (1997) 1426–1434
12. Stok, A., Sargent, E.H.: Lighting the local area: Optical code-division multiple access and quality of service provisioning. IEEE Network **14** (2000) 42–46
13. Ng, E.K.H., Sargent, E.H.: Mining the fibre-optic channel capacity in the local area: maximizing spectral efficiency in multi-wavelength optical CDMA networks. In: IEEE International Conference on Communications (ICC 2001). Volume 3., Helsinki, Finland (2001) 712–715
14. Tancevski, L., Andonovic, I., Tur, M., Budin, J.: Hybrid wavelength hopping/time spreading code division multiple access system. Proceeding of IEE Optoelectronics **143** (1996) 161–166
15. Chang T.W.F and Sargent E.H.: Optical CDMA Using 2-D Codes: The Optimal Single-User Detector IEEE Commun. Letters, Vol. 5. (2001) 169-171
16. Yu, K., Park, N.: Design of new family of two-dimensional wavelength-time spreading codes for optical code division multiple access networks. IEE Electronics Letters **35** (1999) 830–831
17. Chang, T.W.F.: Optical Code-Multiple Access Networks: Quantifying and Achieving the Ultimate Performance. Master's thesis, Graduate Department of Electrical and Computer Engineering, University of Toronto (2000)

Robust Character Image Retrieval Method Using Bipartite Matching and Pseudo-bipartite Matching

Sangyeob Lee[1] and Whoiyul Kim[2]

[1] Management Information System,
Sahmyook University, Seoul, KOREA
zikimi@syu.ac.kr
[2] Division of Electrical and Computer Engineering,
Hanyang University, Seoul, KOREA

Abstract. Studies have been actively undertaken on image retrieval from a large-capacity image database, particularly on image retrievals such as natural image retrieval, character image retrieval or trademark retrieval from a streaming image database. In retrieving a character image, either color or shape information is used for the key feature information. However, changes in the shape of a character image often makes it difficult to retrieve solely based on the shape information, requiring a new retrieval method where both color and shape information are taken into consideration. We present a highly effective method for retrieving or matching similar character images even when shape information differs substantially. In our approach, combined features of color and shape information are used for image retrieval; an image is first split into sectors; color information is extracted from the sector images followed by quantization using Parzen window to extract features; an image is then retrieved by means of bipartite matching using the features. Our results show the retrieval rate using the combined information increases substantially for matching natural or character images compared with the results obtained by the combination of two features independently..

1 Introduction

Diverse studies on how to develop an image retrieval system have been carried out so far. A commonly used method is to index an image's characteristics into strings using DBMS and retrieve an image by means of string query. Among the new methods is one for extracting features from a query image, and comparing them with those of images in a database for retrieval [2,3,4,5,6,7,8,9].The characteristic features generally used to compare a query image with the ones in a database are shape and color [2,3,4,5,6,7,8,9]. Shape has traditionally been used for image retrieval [6,8,9], however in recent years, more studies are being performed with the use of color information [2,3,4,5,7,10,11,12]. When shape is used, shape features of an image object are extracted as a base feature to be coded, or

W. Chang (Ed.): AISA 2002, LNCS 2402, pp. 295–306, 2002.

an object is classified by moment feature like Zernike moment [22]. Shape features are effective to use for an image object whose characteristics are distinct, but they are very difficult to use for an image object that changes diversely, for example, the mickey mouse in animation. Color has recently been used to retrieve such image object as an alternative. Color information is more effective than shape information to use for the object with diverse shape variation because color information rarely changes in such cases. However, an image retrieval method using either color or shape information alone obviously results in poor retrieval accuracy than the one using both information [2], because when only color features are used, a completely different image object with the same color features can be falsely retrieved as being the same. Thus it is necessary to study an image retrieval method that utilizes a combined feature of both color and shape information. If both types of information - color and shape features that are separately extracted from the image - are used, the amount of data becomes enormous. To remedy the problem, studies need to be undertaken on an image retrieval method that optimizes the information by extracting the features in an integrated way. In such an attempt, we tried to split an image object into sector images and then extract color features from each sector image. It proved to be a more effective method than the one using either color or shape information exclusively, and the amount of the data was greatly reduced compared to that produced by the method using both color and shape features that are extracted separately. The simplest and easiest method for extracting color information and indexing it was proposed by Swain [10]. Based on his study, Brian suggested a method for generating a histogram invariant to illumination [11]. The gist of the two methods is to extract color information from an image with the use of color histograms. An image retrieval method using the distance of the information was also developed on the basis of these methods [12]. Since this paper deals with a character image retrieval method that is irrelevant to illumination, we used Swain's method to generate a histogram. But while Swain used the minimum value to compare images, we used distance because we compared them using matching. We also used Parzen window for quantization to minimize the error in extracting features. There are various methods for comparing two images [2,4,8,12,13,14]. Each method has strong and weak points, and has a certain area where it performs well. So far more studies have been performed on how to retrieve an image whose color is changed by illumination than an image whose shape changes diversely [2,4,8,12,13,14]. But in order to retrieve a character image that does not change in illumination but diversely changes in the shape, two image objects need to be compared by the values reflecting the information on their similarity. In this paper, we tried to compare images by means of two graphs of features. The method we used was bipartite matching. The method has mainly been used in the network to find max flow [15,16,20]. It has also been employed in the field of 3D object recognition [1] where the information of an entire image transformed in the form of a tree graph where graph matching method can be utilized for interpretation of the image [13]. In our approach, we split a compared image into sectors, generated a color histogram for each sector,

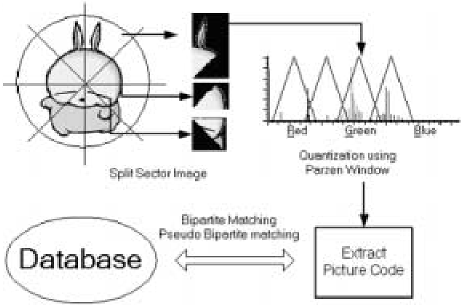


Fig. 1. Char image retrieval system overview

extracted features through quantization using Parzen window, then extracted the distance value from the features using bipartite matching, and evaluated their similarity. Bipartite matching is not efficient in real-time matching because it has basically the complexity of $O\left(n^3\right)$, which requires developing a real-time image retrieval method. We used Pseudo-bipartite matching (PBM), a less accurate but faster method than a bipartite matching method. In Chapter 2, we propose a character image retrieval system, and explain color histogram quantization in Chapter 3, and bipartite matching and Pseudo-bipartite matching in Chapter 4. In Chapter 5, we present the retrieval results of our method conducted with 25 groups composed of 2,000 images using the bipartite matching, PBM and Global Histogram matching, and then conclusions are presented. We expect that our method will be efficiently used to retrieve an image object from streaming images, to retrieve a character image from an image database, or to retrieve an object from real-time images.

2 Overview

The char image retrieval system we suggest in this paper is shown in Fig. 1.

In this paper, we present a method for comparing an object image that has been extracted from split image sectors with one in the database. It works as follows: split an object image into n image sectors; generate color histograms of the n split sectors; quantize the histograms using a triangular Parzen window; and then extract the distance through bipartite matching or PBM with the features in the database. Splitting an image into sectors and generating a histogram of a split sector image means trying to generate a local histogram. While a global histogram gives information about the color distribution of an entire image, a local histogram shows the color distribution of each region. Accordingly, a local histogram includes some information on the shape of an image. A character image often has a very different histogram depending on the region. So for character image retrieval, a local histogram gives more accurate values than a global histogram, and also has the merit of including some shape information. A histogram of a sector image includes a great quantity of information.

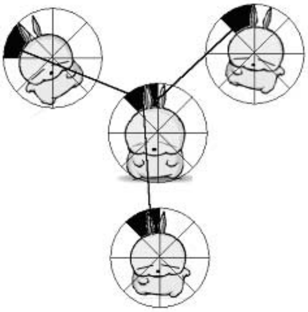


Fig. 2. Using Bipartite matching

When the R,G,B level is 256, a sector image needs 768 bins, and thus a total of 4608 bins is required for 6 split sector images. Therefore, quantization is essential to reduce the quantity. Since a histogram of a character image features peaks in certain areas, quantization does not incur much loss in information. But when peaks lie on the boundary of quantization, a big error is produced. To address this problem, we quantized an image sector using Parzen window with triangular windows overlapped. This way, a smaller error is produced even when peaks lie on the boundary. After extracting features from a local color histogram, we used bipartite matching to compare the values. It is because the method has the merit of adapting well to shape changes and thus ensures stable comparisons. Fig. 2 shows how bipartite matching works for an image sector matching.

The bipartite matching we used in this paper is a weighted bipartite matching method. It works so that weighted values of each node can reach max flow. In Fig. 2, the location of the sector that includes the rabbit's left ear is not the same in each char image. But with a matching method that uses the features extracted from local color histograms as weighted values, sectors in which the color histograms are most similar are matched with one another. Bipartite matching results in an optimized solution, which guarantees an optimized matching in the present picture even though there are a few errors in some parts. Bipartite matching even between the totally different object images, if done with the use of the sector's features, results in the matching between the most similar sectors. In this case, however, the weighted bipartite matching max flow value is far smaller than that produced between the same object images. So if we use the max flow value as distance, it can be a guide for us to evaluate the similarity of two object images. Pseudo-bipartite matching involves the computation of approximate max flow value unlike a bipartite matching that calculates max flow value. The method is less accurate than a common bipartite matching method, but operates very fast when comparing features. If a char image needs to be extracted from a streaming image that is being downloaded on a real-time basis, a very fast method is essential. Because the PBM we suggest here has the complexity of $O\left(2n\right)$, it runs much faster than weighted bipartite matching. While an image database has many similar object images, streaming image data don't, which makes the max flow values differ greatly. Therefore, we opted for PBM for a real-time retrieval.

3 Color Histogram Quantization

We propose a method that splits an object image into sector images, extracts values from the color histogram of each sector, and uses the values as the features of each sector image. But when all the values from the color histograms are used as the features, the volume of the data explodes. Consequently, histogram quantization is essential. But using uniform quantization is problematic because it produces an error when peaks lie on the boundary. A histogram of a char image often features peaks in certain areas, and when the peaks are divided by a boundary, a big error is produced. To fix the problem, we used Parzen window. A general formula for Parzen Window is as follows:

$$r_k = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{x - x_i}{h_n}\right) \tag{1}$$

h_n refers to the length of the quantization region, $x - x_i$ to the distance from the present region, φ to quantization window, and V_n to the volume of φ . (1) is the formula for converting the value of a region that is integrated with the use of φ windows into a feature. In this case, a region's value varies depending on what kind of φ is used. The most idealistic method for minimizing error is Gaussian window. But it requires more computation. Our aim is to minimize error when the peaks representing the features of a char image are on a boundary. In order to reduce computation and minimize error, we used a triangular window. The basic formula used in this study for color histogram quantization is as follows:

$$r_k = \sum\left(\delta_n \times \left[1 - \left(q_k - \frac{h_n}{Q}\right)\right]\right) \tag{2}$$

h_n is the source's histogram level, while q_k is the destination's histogram level. δ_n refers to the value of the source's histogram level, r_k to that of the destination's histogram level, and Q to that of quantization level. Because a char image's histogram values feature a few peaks in certain areas, Formula 2 resulted in the quantization values that didn't differ very much from those calculated using Gaussian window. In order to extract features after the histogram quantization, the values of (2) need indexing. Indexing means that a perfect matching is realized between sector images of two image objects with each edge of the nodes having a weighted value. We used Swain's indexing method for indexing [10]. In Swain's method, the minimum value is used, but we used distance instead.

$$H(I, M) = \frac{\sum_{j=1}^{n}(I_j - M_j)}{\sum_{j=1}^{n} M_j} \tag{3}$$

$H(I, M)$is histogram indexing, I_j a quantized histogram value of a query image, and M_j a quantized histogram value of a database image. The difference between

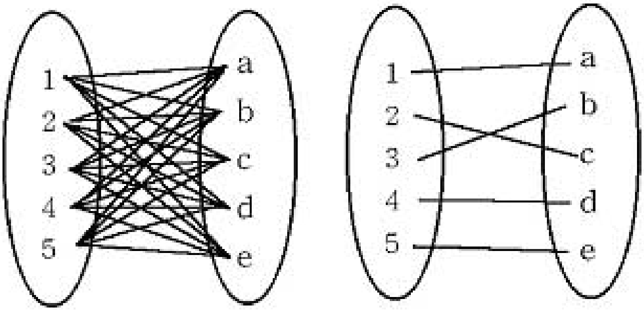


Fig. 3. Perfect matching and complete matching

the two images is presented as a ratio because the total sum of the differences between the histogram values of the two images was divided by the sum of the database images' histogram values. When indexing is finished, a perfect matching is completed in a bipartite graph. And a method for finding max flow helps the matching of each sector image.

4 Weighted Bipartite Matching

Weighted bipartite matching refers to a complete matching in which each edge is established in a bipartite graph so that max flow can be reached. Bipartite matching is explained in detail in [1,17,18,19,20]. We intend in this paper to explain its overall idea. Fig. 3 illustrates a complete matching and a perfect matching of weighted bipartite matching.

When the nodes of Graph G are grouped as V and V1, it is called a bipartite graph, and when each node u of V is connected with each node V of V1 as an edge, a perfect matching is realized. Under the perfect matching condition, $c(u.v) > w(M)$. M refers to G's perfect matching. If $c(u,v) > w(M)$, then $ui + vj = wij$. At the point of $max(\sum wij)$, maximum weight matching is realized. Maximum weight bipartite matching is generally called bipartite matching, and that is when complete matching is realized.

4.1 Maximum Weight Bipartite Matching

A principal algorithm for maximum weight bipartite matching is the Hungarian method [1,18,19].It basically starts with the idea that subtraction of the minimum value from each edge would not influence the optimized flow. In other words, when the matching is maintained to form G's subgraph in which the edge cost is minimized, a perfect matching is realized. The Hungarian method basically has the complexity of $O\left(n^3\right)$ [18,19]. The following is the formula for computing the total cost when an arbitrary edge pair is established while maintaining a maximum weight bipartite matching.

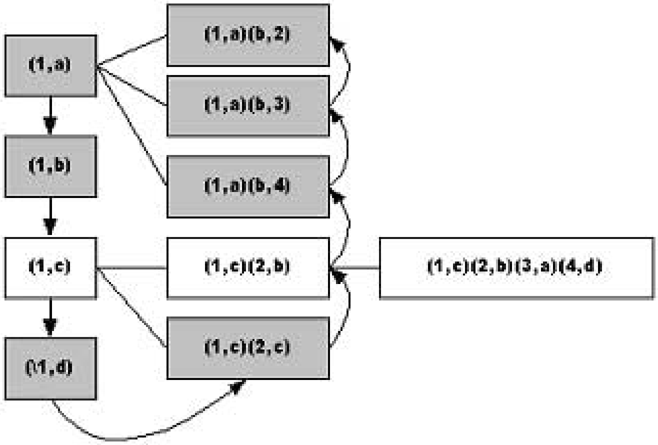


Fig. 4. Illustration of a sequence search that selects the node pairs that cannot increase nodes

$$E = \sum E_{xy} + \varphi_{x'y'} + \sum_{j=1}^{n} min(w_{ij}) \tag{4}$$

In Formula (4), E refers to the total cost of the edge pair that is presently established, and E_{xy} to the cost of the edge that is presently under maximum matching. $\varphi_{x'y'}$signifies the cost of the edge that is newly proposed for maximum matching, while $min(w_{ij})$ the minimum value of the potential paths to be connected among the nodes that have yet to be matched. As is shown in Formula (4), maximum weight bipartite matching is a method in which the algorithm tries to find the edge pairs that minimize the cost while increasing the number of repetitive nodes, and calculates the E value. And in the process, the algorithm dismisses the matching pairs as dead paths when the previous E value exceeds the present one. A common method for maximum matching is to use an alternative tree, in which max flow is reached while the algorithm repeatedly eliminates the node pairs in the tree that can no longer generate leaves. A circulative method is generally used to increase the number of eligible augment paths. In this paper we used sequence search to remove node pairs. Fig. 4 illustrates how sequence search works. The gray-colored node pairs in Fig. 4 are those that cannot add nodes. In the node pair search, a linked list search is available without having to use a tree search by also linking the list pairs with arrows in the existing tree structure. When trying to pick out a node pair of the tree that cannot generate leaves, a tree search requires circulating all the previous steps while moving to the root from the position where the current augment path is located, and then moving again to the leaves. But linked lists are much simpler to use than tree leaves, and require less computation. Fig. 5 portrays the structure of an object that stores search data.

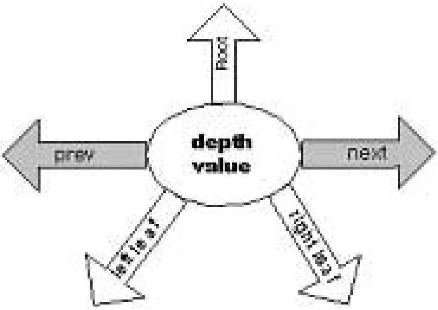


Fig. 5. Structure of an object storing search data

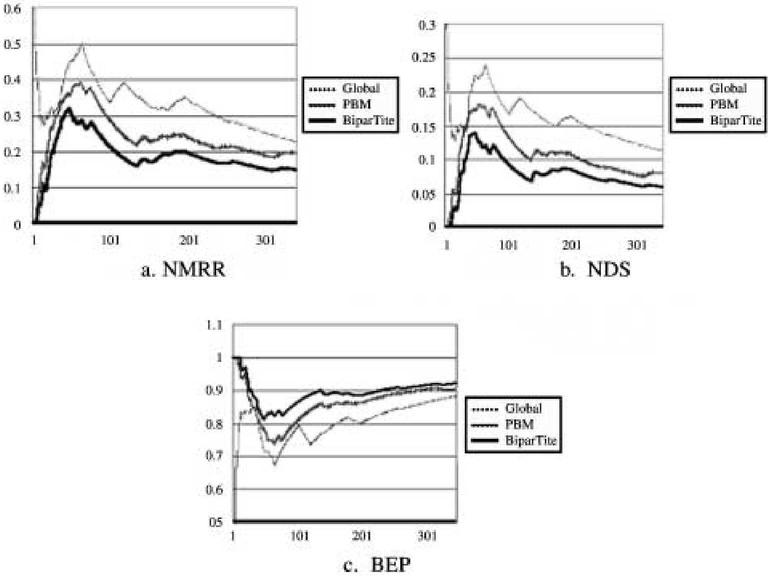


Fig. 6. Graphs of Test Results

In Fig. 5, the gray-colored arrows ‘prev’ and ‘next’ are the edges used for searching the node pair that can’t increase the number, while ‘root’, ‘left leaf’ and ‘right leaf’ are the edges that enable the tree to grow. Under the structure shown in Fig. 6, the steps for augment path increase are followed differently from the search steps.

4.2 Pseudo-maximum Weight Bipartite Matching (PBM)

Because bipartite matching, however improved its speed may be, has basically the complexity of $O(n^2)$ or above, it is not efficient for real-time image retrieval from a streaming image database. We introduce Pseudo-bipartite match-

Table 1. Test Results

	GHM	WBM	PBM
NMRR	0.1133	0.0602	0.0795
NDS	0.2279	0.1504	0.1951
BEP	0.8825	0.9226	0.9031

GHM: Global Histogram Matching
WBM: weight bipartite matching
PBM: Pseudo-bipartite matching

ing (PBM), which calculates an approximate max flow. PBM's basic approach is to extract max weighted values in due sequence from the Graph M that realizes a complete matching while trying to induce a perfect matching, to subtract the minimum matching weighted value, and to finally compute an average flow for each node pair.

$$M_k = max(w_{ij}) \tag{5}$$

$$E = \frac{(\sum M_k - min(M_k))}{n-1} \tag{6}$$

(5) explains that the algorithm connects the nodes that have max weighted values under the perfect matching condition, completing the matching of the connected nodes, and once again tries to find max weighted value. When n edges are connected by (5), the smallest weighted value is subtracted from the total sum of the weighted values of the n edges, and then the result is divided by $n - 1$, producing an average matching value of (n -1) edges. (6) is a formula for calculating an average matching value E. Our concern in image matching is about how big the distance between the query and model images is, and consequently we need to know the global relationship between them, not the relation between each node pair. In a PBM approach, the matching with the greatest distance has the value of $min(M_k)$. To disregard it means to disregard the value of that part of a changing image with the most dramatic change. In matching images whose shapes greatly change, elimination of the part that changes most actually decreases the distance value, working in favor of extracting the same image. Actual experiments using the above approach result in an approximate max flow. Because of its low complexity ($O(2n)$), PBM method runs very fast. Thus PBM method proves to be efficient for a real-time image matching system, though it is a little less accurate.

5 Conclusion

In order to evaluate the method proposed in this paper, comparative experiments were carried out with Swain's Global Histogram Matching that generates a global histogram by color indexing, and then calculates the distance to measure the similarity, and the performance was evaluated using BEP(Bull's Eye Performance), NMRR(Normalized Modified Retrieval Rank), NDS(Normalized

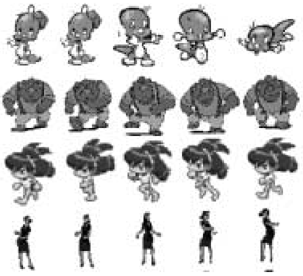


Fig. 7. Char sets where bipartite, Global and PBM all performs well

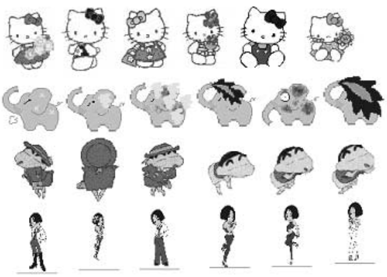


Fig. 8. Char sets where bipartite matching and PBM performs well

Distance Sum). The method for recognizing char image shapes was compared with the method that uses the Zernike moment. But the Zernike moment method didn't work very well for comparison, so we opted to compare it with Global Histogram Matching. BEP measures the percentage of the related images actually searched by counting the number of the related images that are doubled in number. The following is the formula for BEP.

$$BEP(\%) = \frac{R_{s2r}}{R} \times 100 \tag{7}$$

R_{s2r} is the number of related images that are doubled in number. The closer to 1 the BEP, the better the performance. NMRR is a measure used by the Color/Texture descriptor group of the MPEG-7 Standardization Conference. When measured by NMRR, the closer to O the result, the better the performance. NDS is a measure proposed in [21]. The closer to O the NDS, the better the performance. NDS's strong point is that it considers every condition, which makes fine measuring possible. Table 1 is the results of Global Histogram Matching, Bipartite Matching and PBM.

For the test, a total of 2,350 images were classified into 25 char image groups, and 342 images from the groups were used. The best method in performance is bipartite matching, which attained the best results in all evaluations, followed by PBM, and GHM. PBM has poorer performance than bipartite matching, but its flow is similar to that of Bipartite Matching. Fig. 6 portrays the test results in graphs. Global Histogram Matching performed very well in certain groups. Fig. 7 shows some char sets where GHM, PBM and bipartite matching all perform well. Global Histogram Matching performed well because each char set in the char group has the same color histogram distribution. But when some of the images of a char set have a big difference in their global histograms, GHM produces a big error. Fig. 8 shows some of the char sets where GHM performs poorly but Bipartite matching and PBM performs well.

As is shown in Fig. 8, some of the images in the same char set have partially different color values. In this case, GHM produced a high dissimilarity value while bipartite matching or PBM produced a small one. It is because an image is segmented into sectors for bipartite matching or PBM, and so even if the distance between a certain sector image is big, the distance value remains small due to the maximum matching. It is very difficult to extract an object image when trying to retrieve one from the streaming images, noises are made by diverse changes. We think such a problem can easily be solved with the use of our method.

References

1. W. Y. Kim and C. Kak: 3-D Object Recognition Using Bipartite Matching Embedded in Discrete Relaxation, IEEE Trans. Pattern Anal. Machine Intell., **13** (Mar. 1991) 224–251
2. T. Gevers and A. W.M. Smeulders: PicToSeek: Combing Color and Shape Invariant Features for Image Retrieval, IEEE Trans. Image Proc., **9** (Jan. 2000) 102–119
3. C. S. Fuh, S. W. Cho, and K. Essig: Hierarchical Color Image Region Segmentation for Content-Based Image Retrieval System, IEEE Trans. Image Proc., **9** (Jan. 2000) 156–162
4. N. Vasconcelos and A. Lippman: Featrue Representations for Image Retrieval: Beyond The Color Histogram, IEEE Int. Conf. ICME 2000, **2** (2000) 899–902
5. N. Sebe and M. S. Lew: Color Based Retrieval and Recognition, IEEE Int. Conf. ICME, **1** (2000) 311–314
6. T. K. Shih and C. S. Wang: Indexing and Retrieval Schme of the Image Database Based on Color and Spatial Relations, IEEE Int. Conf. ICME , **1** (2000) 129–132
7. A. Mojsilovic: A Method For Color Content Matching Of Images, IEEE Int. Conf. ICME 2000, **2** (2000) 899–902
8. R. Mehrotra and J. E. Gary: Similar-Shape Retrieval In Shape Data Management, Computer, **28** (Sep. 1995) 7–14
9. S. Sclaroff and A. P. Pentland: Search by Shape Examples: Modeling Nonrigid Deformation, Signals, Systems and Computers, **2** (1994) 1341–1344
10. M.J. Swain and D.H. Ballard: Color indexing, Int'l. j. Comput. Vision, **7** (1991) 11–32
11. B. V. Funt and G. D. Finlayson: Color Constant Color Indexing, IEEE Trans. Pattern Anal. Machine Intell., **17** (May 1995) 522–529
12. J. Hafner and H. S. Sawhney: Efficient Color Histogram Indexing for Quadratic Form Distance Functions, IEEE Trans. Pattern Anal. Machine Intell., **17** (Jul. 1995) 729–736

13. A. D. J. Coross and E. R. Hancock: Graph Matching With a Dual-Step EM Algorithm, IEEE Trans. Pattern Anal. Machine Intell., **20** (Nov. 1998) 1236–1253
14. D. Shasha, J. Tsong-Li Wang, K. Zhang and F. Y. Shih: Exact and Approximate Algorithms for Unordered Tree Matching, IEEE Trans. System, Man, and Cybernetics, **24** (Apr. 1994) 668–678
15. M. Goldstein, N. Toomarian and J. Barhen: A Comparison Study of Optimization Methods for The Bipartite Matching Problem (BMP), IEEE Int. Conf. Neural Networks, **2** (1988) 267–273
16. E. Cohen: Approximate max flow on small depth networks, Foundations of computer Science, Proceedings, 33rd Annual Symposium on, (1992) 648-658
17. K. P. Bogart: Introductory Combinatorics, A Harcourt Science and Technology Company, (2000) 291–358
18. D. B. West: Introduction to Graph Theory, Prentice Hall, (1996) 98–132
19. H. W. Huhn: The Hungarian method for the assignment problem, Noval Research Logistics Quartely, (1995) 83–97
20. P. M. Vaidya: Geometry helps in matching, Proceed. Twentieth Annual ACM Symposium on Theory of Computing, (1988) 422–425
21. C. Seo and W. Kim: Content-based similar-trademark retrieval from an image database and a new evaluation measure of retrieval efficiency, Journal of the Korean Society of Broadcast Engineers, **5** (2000) 68–81
22. Y. Kim and W. Kim: A region-based descriptor using Zernike moments, Signal Processing: Image Communication, **16** (Sep. 2000) 95–102

Author Index