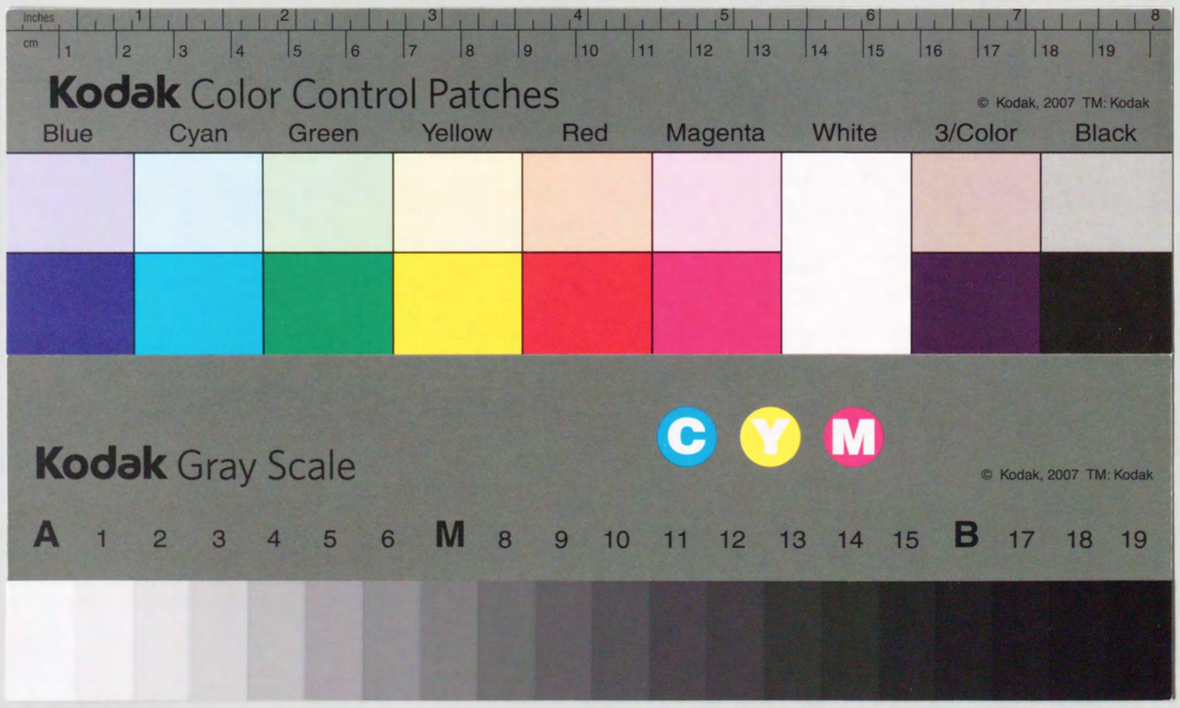
Studies on Dynamic Admission and Flow Control in ATM Networks

Kohei Shiomoto



Studies on Dynamic Admission and Flow Control in ATM Networks

Kohei Shiomoto

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Engineering
(Informatoin and Computer Sciences)
at
Osaka University
Toyonaka, Osaka, Japan
February 1998

Abstract

In the asynchronous transfer mode (ATM), in which the transmitted data is segmented into fixed-sized packets called *cells*. These cells are directed towards their destination by switching fabric hardware; the cell transmission rate is adjustable. Thus, ATM is considered a promising technology for future broadband integrated services digital networks (B-ISDNs), which will handle multimedia traffic.

The telecommunications industry is changing rapidly. Many multimedia applications are being developed, and traffic volume is growing exponentially, as seen in the recent explosive growth of the Internet. To handle this rapid growth and change, future networks must be scalable and adaptive. This dissertation focuses on the scalability and adaptability of ATM networks.

The switching nodes in an ATM network must be easy to scale to handle the growing traffic volume expected in a mature multimedia networks. The second chapter of this dissertation we describe a scalable ATM switch architecture. Multi-stage switching networks are used to scale the ATM switching fabric. A multi-stage switching network is composed of a number of regularly interconnected switching elements. A Banyan network is an example of a multi-stage network. As the size of the switching network increases, the maximum throughput is limited by internal blocking and head-of-line blocking. To remove this throughput limitation, we studied a cell-bypass queueing mechanism in which subsequent cells can be bypassed when both head-of-line cells have the same destination. A key design parameter is to what extent cells can be bypassed. To set this parameter, we derived such performance measures as cell loss probability, mean delay time, and the maximum throughput by analyzing a discrete Markov chain for both uniform and non-uniform traffic patterns. Our analysis showed that the cell bypass queueing discipline greatly improves Banyan network performance with respect to both throughput and cell blocking probability, while keeping fairness under non-uniform traffic patterns. A Banyan network with a cell-bypass queueing discipline enables us to make an ATM switch to be scaled while maintaining throughput. We can thus build a scalable ATM switching node to meet the rapid growth in traffic volume expected in future B-ISDNs.

Traditionally, a traffic engineering formula has been constructed every time a new application or service is introduced. Presently, however, many kinds of applications are being developed at an unprecendented rate. In the third chapter we introduce a measurement-based admission control for making ATM traffic control adaptive. Analysis of the cell loss ratio as a function of buffer size showed that burst-level traffic fluctuation is crucial for admission control. We thus developed a method of designing a low-pass filter that estimates the instantaneous rate, defined as the sum of the peak rates of the active connections, and then developed an admission control method in which the maximum instantaneous rate is used to derive the residual bandwidth. The monitoring period is set such that the cell loss ratio is maintained. Our evaluation of the performance of this admission control method by queueing analysis and simulation techniques showed that it achieves high

bandwidth efficiency. An example implementation showed that this method can be implemented using simple hardware. Because this method is based on real-time traffic measurement and does not assume any kind of traffic model, it is suitable for traffic with unpredictable characteristics.

Various types of multimedia traffic will be handled in ATM networks, so these networks must be able to accommodate different traffic types, i.e., quality of service (QoS) targets and traffic characteristics. An efficient way is thus needed to handle traffic with different QoS-characteristics. In the fourth chapter of this dissertation we present a simple and efficient adaptive admission control method for handling multiple QoS targets, specifically, the cell loss ratio (CLR). The proposed method alternates a separate control using multiple queues and an aggregate control using an individual CLR. The decision as to which control should be used is based on the number of connections, i.e., if the weak-burst type connections request a higher QoS (a lower CLR) than the other connections and if they are a small fraction of the total traffic volume, separate control is used. Otherwise, aggregate control is used. By using this adaptive admission decision strategy we can maximize the number of admitted connections. For separate control, we use a binary-search algorithm and the history of the number of admitted connections to allocate the bandwidth to each queue so as to reduce the time needed to re-allocate the bandwidth when a new connection is admitted or an existing one departs. For aggregate control we introduce two fast algorithms for calculating the individual CLRs. Because the proposed connection admission control method is simple and fast, it achieves high bandwidth efficiency in a multiple QoS environment. It is thus suitable for future multimedia ATM networks.

The role of admission control is to maintain an extremely low target CLR. Admission control is *preventive* control, so we need to determine whether the admission control algorithm works properly. The target value of CLR is considerably low, e.g., 1.0e-5 to 1.0e-8. It is thus difficult to determine whether the target CLR is maintained or not. In the fifth chapter of this dissertation we present a method for estimating the CLR. We have developed a real-time algorithm for estimating the Markov modulated Poission process parameters used to calculate the CLR. The developed algorithm uses a fictitious queue and a window to determine the load states, i.e., underload or overload. We evaluated the performance of this algorithm through computer simulation and found that the CLR can be estimated closely by setting the window size to the minimum cell interval of the connection. The proposed algorithm is thus suitable for admission control. Computer simulation confirmed that admission control based on this algorithm achieves high bandwidth efficiency.

Acknowledgments

I would like to express my sincere appreciation to Prof. Hideo Miyahara of Osaka University, my adviser, for his countless suggestions and continuous support. His courses on communication networks and queueing theory opened my eyes to these fields. Without him, I would not be where I am today. I am heartily grateful to Prof. Akihiro Hashimoto and Prof. Kenichi Taniguchi for serving as readers on my dissertation committee. Their expertise and insightful comments have been very beneficial.

I would like to express my gratitude to Prof. Masayuki Murata for instructing me when I was a graduate student at Osaka University. He showed me how interesting it is to evaluate the performance of networking systems. His active and enthusiastic approach to research serves as a model for my efforts. Without his guidance, I would not have been attracted to this wonderful world of research. Throughout my work on this dissertation he provided many helpful comments and corrections. Without him, I could not have produced this dissertation.

Sincere appreciation is due to Prof. Yuji Oie of Kyusyu Institute Technology for his valuable comments and suggestions on the material in Chapter 2, which was developed when I was a graduate student, and to Prof. Syoujiro Nishio, Prof. Motohide Umano, Dr. Takeshi Nishida (currently, NEC corporation), Prof. Shinji Shiomojo, Prof. Suguru Yamaguchi (currently, Nara Institute of Science and Technology), Prof. Masashi Sugano (currently, Osaka Prefectural College of Health Sciences), Mr. Masaaki Hyoudou (currently, Sharp corporation), and other members of Osaka University for their comments and suggestions they gave me when I was a graduate student of Osaka university.

This dissertation would not have also been possible without the support of Dr. Shigehiko Suzuki of NTT Network Service Systems Laboratories. I express my gratitude to him for giving me the opportunity to write this dissertation.

The materials in Chapter 3, 4, and 5 were generated since I joined the NTT laboratories. I am greatly indebted to Prof. Hiromasa Ikeda of Osaka University (Former Director of the NTT Communication Switching Laboratories), Dr. Toshiharu Aoki, Dr. Hiroshi Ishikawa, Mr. Tohru Uchiyama (currently NTT Advanced Technology coporation), Mr. Takeo Koinuma (currently NTT Electronics corporation), Mr. Hirokazu Ohnishi, Dr. Tatsuro Takahashi, Dr. Naohisa Ohta, Dr. Toyofumi Takenaka, Mr. Zen-ichi Yashiro, and Dr. Noriharu Miyaho for their guidance and encouragement.

Special thanks are due to Dr. Naoaki Yamanaka, who continuously gave me his guidance and encouragement while I was writing this dissertation. I would like to express my sincere appreciation to Dr. Yoshitaka Takahashi, Dr. Hiroshi Saito, Mr. Kiyohiro Noguchi, Ms. Miki Hirano, Mr. Shin-ichiro Chaki, Mr. Haruhisa Hasegawa, Mr. Jun Itoda, Mr. Eiji Oki, and other researchers of NTT laboratories for their comments and discussions related to this dissertation. Some of the ideas presented in this dissertation were developed when I was engaged

in development of first commercial ATM switching system. Thanks are due to the researchers who worked with me on that project: Mr. Ken-ichi Kuroda, Mr. Kouichi Seino, Dr. Kou Miyake, Mr. Yukihiro Doi, Mr. Yasushi Takagi, Mr. Arata Itoh, Mr. Kazukuni Koyama, and other members.

I would like to express my heartfelt appreciation to Prof. Jonathan S. Turner for his guidance when I was staying at Washington University in St. Louis, Missouri, U.S.A. Without his insightful suggestions, my research there would not have been successful. I also thank Mr. Qiyong Bian for our most enjoyable discussions while I was there.

I would like to express my sincere appreciation to my father Kazumi. Not only did he work hard to raise me, he always supported me. From the time I began struggling with my homework in elementary school, he taught me how important it is to solve problems systematically. I am also sincerely grateful to my mother Toshiko for her support and encouragement. She worked hard to develop in me the habit of being positive about everything. My elder sister Taeko also played an important role in my development by taking care of me when I was a child.

Finally, I would like to express my thanks and my love to my wife Yuko and my daughter Ayano. Their smiles comfort me and give me energy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we describe key concepts addressed in this dissertation: asynchronous transfer mode (ATM), ATM switching architecture, and connection admission control. First we introduce ATM which brings us the broadband integrated services digital networks in Chapter 1.1. Next we introduce an ATM switching architecture in Chapter 1.2. We then describe the connection admission control to meet various QoS requirements in the multimedia B-ISDN in Chapter 1.3. An outline of this dissertation is shown in Chapter 1.4.

1.1 Asynchronous Transfer Mode

The development of computer technology enables us to enjoy the advantages of digitized information. Information represented in digital form can be easily manipulated, reproduced, and transmitted [2]. Many kinds of information can be digitized, such as voice, video, text, computer program data, and so on. There is strong motivation to handle these different kinds of information in a unified way. Computer networks play an important role in transmitting digitized information. In the multimedia environment, information rate and holding period range widely. Telephony and hi-fi video stream services require low and high bandwidth for a considerable period, while data transfer requires high bandwidth but lasts only for a short time. Traditionally, telecommunication networks are developed separately so as to optimize individual services. However, a separate network means high cost: development, equipment, operation, and maintenance. In addition, demand for each service is not easy to forecast. The expectation is that these issues will be dealt with by broadband integrated service digital networks (B-ISDN).

Asynchronous transfer mode (ATM) is a connection-oriented communication paradigm, in which all the information is segmented into fixed-sized packets called *cells* [3]. Cells are switched by switching fabric hardware and the cell transmission rate is adjustable. ATM is thus considered to be a promising technology for future broadband integrated service digital networks (B-ISDN) for handling multimedia traffic. ATM has been standardized as the infrastructure of B-ISDN by two standardization bodies: the International Telecommunication Union-Telecommunication Standardization Sector (ITU-TS), formerly the Consultative Committee on International Telephone and Telegraph (CCITT) and ATM Forum.

The main feature of ATM includes self-routing switching and label multiplexing. Self-routing switching is suitable for hardware implementation. It enables cells to be switched to the proper output port by routing

tags attached to the proprietary cell header. Regardless of which input port the cells are injected into, they are switched to the output port whose identifier matches the routing tag. In this sense, self-routing switching is a passive device. Fixed-sized cells and self-routing mechanisms suit hardware implementation. Hardware implementation allows us to make the switching fabric scalable by regularly inter-connecting the switching elements.

In ATM networks, a cell is identified by the label attached to its header [4]. The routing decision is performed at connection set-up. The state information on association of a label and a routing tag is stored at an input port unit of the switching fabric at connection setup. The label attached to the cell header is used to derive the routing tag necessary to switch the cell. Unlike circuit switching, in which the time slot for each communication is synchronized throughout the network, ATM enables us to send cells at any time by virtue of the label multiplexing concept. We can adjust the cell transmission rate: when we have no information, we don't have to send cells and can leave the bandwidth to others, thus realizing statistical multiplexing. Since ATM is connection-oriented, it is suitable for implementing a bandwidth guarantee mechanism. When a connection is set up, it is checked to determine whether or not sufficient bandwidth is available to maintain certain levels of Quality of Service (QoS). This mechanism is called admission control, and is one of the main themes of this dissertation.

1.2 ATM switching architecture

As the multimedia network infrastructure progresses toward the next millennium, the network should be expanded to handle subscriber rates of up to Mega bits per second from the current 64 Kbit/s telephony. Accordingly the switching nodes should scale up to Tbit/s switching capacity to handle future multimedia subscribers. In future multimedia networks, the bandwidth requirements will range widely. The switching nodes should also scale according to growth of traffic demand.

The self routing ATM switch fabric is an essential component of ATM networks. The routing tag attached to the cell header dictates the destination port. When several cells simultaneously contend for the same output port, they are queued at input ports or output ports. An input-buffer switch stores those contending cells at an input buffer other than the one selected to be sent to the output port, while an output-buffer switch stores all the contending cells at the output buffer. An input-buffer switch requires an arbitration mechanism to choose the cell to be served from among the contending cells within a single cell slot time. Since there are potentially N contending cells, the arbitration mechanism should be faster than N times the input and output port rate. However, an output-buffer switch requires a buffer whose access time is fast enough to simultaneously store N cells within a single cell slot. The arbitration mechanism is implemented in wired logic, while the buffer is implemented in FIFO memory. Since it is difficult to implement a high speed access time FIFO memory as the access speed increases, an input buffer type ATM switch has been used to construct a high speed core switching fabric [5].

Even though an input buffer switch is suitable for a high-speed core ATM switch, it is difficult to arbitrate N input ports within a single cell transmission time as the number of input ports increases [5]. A large number of multi-stage interconnection networks have been proposed [6] to solve this problem. Units of switching fabrics (we call them switching elements) are interconnected with each other in a multistage fashion. The Banyan network is a multistage switching network. Fig 1.1 shows the 16×16 Banyan network with 2×2 switching

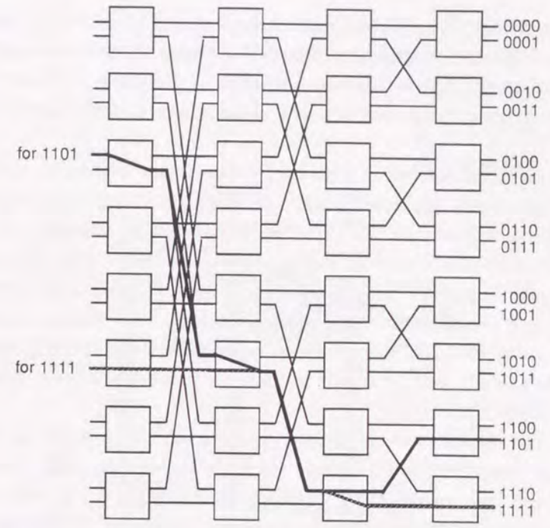


Figure 1.1: 16×16 Banyan network.

elements (SEs). Each switching element is linked to those at its adjacent stages such that a unique path can be established from every input trunk to every output trunk. A cell at stage n is automatically transferred to stage $n+1$ according to the n-th bit of the destination address of the output port. If the corresponding bit is "0", the cell is passed to the upper output port, while in the other case (the bit is "1"), it is passed to the lower output port. A multistage network allows us to implement an ATM switching system with small to large capacity according to traffic demand.

As the size of the Banyan network increases, the maximum throughput is limited by head-of-line (HOL) blocking. HOL blocking refers to the following phenomenon. If cells at the heads of buffers on both input ports are addressed to the same output port in a switching element, one of them is blocked. The other output port is thus left unused during the clock cycle, even if one of the cells remaining in the buffer is destined for that unused output port.

Chapter 2 considers a non–FIFO queueing discipline to eliminate the influence of HOL blocking. We call this the cell bypass queueing discipline; subsequent cells can be bypassed when both head-of-line cells are destined for the same output. Through analysis and simulation, we investigate the effect of a key design parameter of this discipline: to what extent the cells should be bypassed.

1.3 Connection admission control in ATM networks

Since ATM is connection-oriented, we can implement a mechanism for guaranteeing the bandwidth. When the connection is set up, it is checked to determine whether or not sufficient bandwidth is available. Since an

ATM network is a high speed network, rate-based traffic control should be used rather than credit-based traffic control [7]. That is, network resource management is performed using a traffic descriptor for cell transmission rate. Connection admission control (CAC) determines whether or not a new connection setup request should be accepted by taking into account the current network resource status, traffic descriptor and target QoS for individual connections [8, 9].

Figure 1.2 illustrates the connection setup sequence. When a new connection setup request is invoked, signaling messages including information elements (IEs) such as traffic descriptor and target QoS level for the connection are transmitted to the network [10]. CAC is then performed at each node. If the connection is admitted at the node, the bandwidth is reserved and the message is passed to the downstream node.

Of the QoS measures, cell loss ratio (CLR) and cell transfer delay (CTD) are affected by bandwidth utilization. Therefore, they are QoS measures that CAC should consider. Since the maximum CTD is determined by buffer size and link capacity, the CLR has been used as a QoS measure [11, 12], and is also considered in this dissertation. The role of CAC in CTD is how to limit the number of hops in determining the route by constraining end-to-end target CTD.

As mentioned earlier, CAC is based on a traffic descriptor for cell transmission rate. A user should declare peak and average rates at connection setup. These rates are checked by usage parameter control (UPC) to determine whether or not they exceed those declared values (Fig. 1.3). If they do, they are discarded immediately or marked for low priority treatment in a congestion situation. Since it is difficult to accurately estimate these rates beforehand, the declared values may need to include a safety margin to avoid being discarded or marked by UPC, which leads to low bandwidth efficiency. Chapter 3 studies measurement-based admission control as a means of solving this problem. We explore a low-pass filter design for estimating the instantaneous rate and present an admission control method based on measured maximum instantaneous rate.

Since ATM networks must handle multi media traffic, they should accommodate different traffic types with respect to target QoS and traffic characteristics. We need to determine an efficient way to handle multiple QoS and traffic characteristics. In Chapter 4, we study an adaptive admission control method for multiple target QoS levels. A proposed CAC method, in which separate and aggregate controls are selected adaptively, achieves high bandwidth efficiency for any combination of QoS and traffic characteristics.

Since admission control is *preventive* control, we need to validate whether the admission control algorithm actually works properly. Since one role of CAC is to maintain the target CLR, we need to validate whether or not the actual CLR is lower than the target value, which is extremely low, e.g., 1.0e-5 to 1.0e-8. Thus, it is hard to judge whether or not the target QoS level is maintained. Chapter 5 presents a real-time MMPP parameter estimating method for calculating CLR.

1.4 Outline of this dissertation

Change in the telecommunications industry is rapid. Many multimedia applications are being developed and traffic volume is growing exponentially, as seen in the recent explosive growth of the Internet [13]. Because of this rapid growth and change, future networks should be scalable and adaptive. This dissertation focuses on the scalability and adaptability of ATM networks. The following chapters address scalable ATM switch architectures, measurement-based adaptive admission control for traffic whose characteristics are unpredictable, an adaptive admission control for multiple QoS levels, and a real-time algorithm for estimating CLR. The road

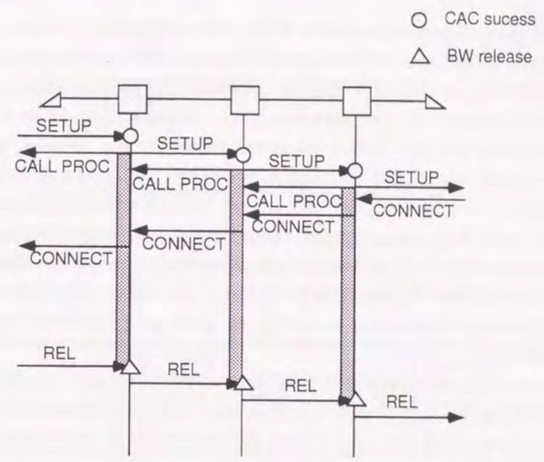


Figure 1.2: Connection setup procedure.

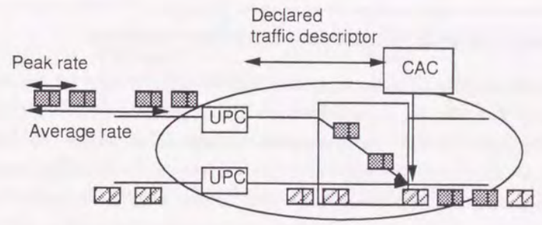


Figure 1.3: Rate-base traffic control in ATM networks: UPC and CAC.

map of this dissertation is shown in Fig 1.4.

1.4.1 A scalable ATM switching architecture

The first part of this dissertation addresses a scalable ATM switch architecture [14, 15], in particular, Banyan networks, which use 2×2 input-buffer type switches as switching elements. As discussed earlier, the throughput of such a Banyan network is limited by HOL blocking. It was already shown that a Banyan network composed of 2×2 switching elements with a single buffer attains a maximum throughput of around 0.45. Even with a FIFO multiple buffer at each switching element, only a minor advantage can be obtained over the single-buffered Banyan network. This is because head of line blocking occurs in the FIFO multiple buffer.

To eliminate the influence of such head of line blocking, we consider a non-FIFO queueing discipline at the two input buffers of the 2×2 switching element, called the *cell bypass queueing discipline*, in which subsequent cells can bypass the head-of-line cell if both of head-of-line cells are destined for the same output. Head of line blocking can be eliminated as follows: if the cell at the head of the input buffer loses an output contention, the following cells within the range of search (or scanning) are allowed, in succession, to contend for an output port until one of them is successful.

A key design parameter for this discipline is how far cells can be bypassed. We present an exact analysis for the 2×2 switching element with input buffers based on this discipline. We derived performance measures such as cell loss probability, mean delay time, and maximum throughput by analyzing a discrete Markov chain for both uniform and non-uniform traffic patterns. Furthermore, an approximate analysis is developed for the Banyan network, composed of 2×2 switching elements with a cell bypass queueing discipline. We investigate the effect of that discipline on performance through numerical results obtained from our analyses as well as from simulation results. We also examine nonuniform input traffic models. Our analysis shows that the cell bypass queueing discipline greatly improves the performance of the Banyan network with respect to both throughput and cell blocking probability, while keeping fairness under non-uniform traffic pattern.

Since a Banyan network with a cell-bypass queueing discipline enables us to make an ATM switch scale well while maintaining throughput, we can build a scalable ATM switching node to meet the rapid growth in traffic volume in the future B-ISDN.

1.4.2 A measurement-based adaptive admission control

Traditionally, traffic engineering plays an important role in provisioning a network resource. When a new service or application is developed, cell traffic is characterized, and mathematical models are constructed. Operational formulae are derived on the basis of these mathematical models. In the future multimedia network, various kinds of traffic are mixed, leading to diversified traffic characteristics. In addition, many kinds of applications have recently been developed at an unprecedented rate. Under such a diversified environment and rapid changes of traffic characteristics, it is hard to establish mathematical models for individual traffic types, and to dimension the network resource. In addition, if we take the above approach, every time a new application emerges, we will need to establish a new mathematical model. To solve this problem, we consider an adaptive admission control based on real-time traffic measurement. In the second part of this dissertation, we address a measurement-based admission control [16, 17, 18, 19]. First, we analyze the cell loss ratio as a function of buffer size and show that burst-level traffic fluctuation is crucial to admission control. Then we develop a method for

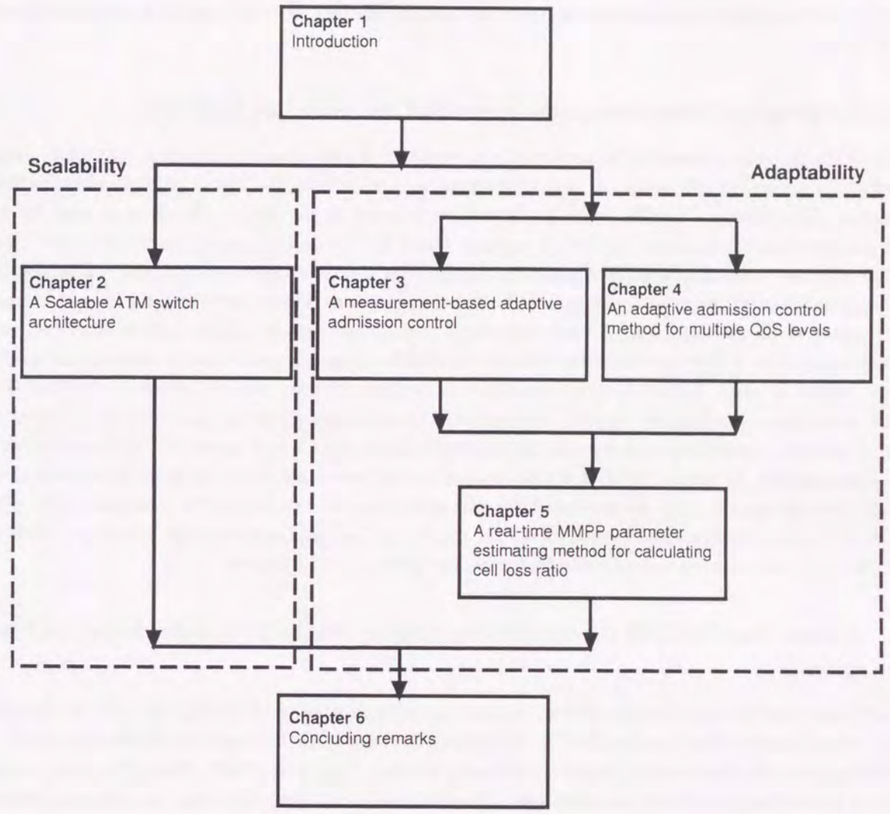


Figure 1.4: Road map of the dissertation.

designing a low-pass filter to estimate an instantaneous rate defined as a sum of peak rate of active connections sending a burst. We also developed an admission control method, in which the maximum instantaneous rate is used to derive the residual bandwidth. We analyzed the monitoring period to maintain a certain cell loss ratio by solving a queueing model. We derived a distribution for the number of connections admitted by the proposed admission control method through queueing analysis and a simulation technique. We also addressed implementation of the method and showed that it can be implemented by simple hardware. Since the method is based on real-time traffic measurement and does not assume any kind of traffic model, it is suitable for the future B-ISDN.

1.4.3 An adaptive admission control method for multiple QoS level

Since the ATM network is intended for multimedia networks, it should support diversified QoS requirements and traffic characteristics efficiently. A new traffic type may appear in the course of multimedia network development. The number of traffic types is expected to increase in the future. We present a simple and efficient adaptive admission control method for multiple target QoS levels with respect to cell loss ratio (CLR) [17]. The proposed method alternates a separate control using multiple queues to separate the bandwidth, and an aggregate control using an individual CLR. The decision as to which control should be used is based on the number of connections., i.e., if weak burst type connections request higher QoS (lower CLR) than other connections and if they are small fractions of total traffic volume, separate control is used. Otherwise, aggregate control is used. With this adaptive admission decision strategy, we can maximize the number of admitted connections. For separate control, we employed a binary-search algorithm and used the history of the number of admitted connections to determine the bandwidth allocation for each queue to improve a bandwidth re-allocation time for the queue, in which a new connection has been admitted or an existing connection has departed. For aggregate control, we developed two fast algorithms for calculating the individual CLR. Since the proposed connection admission control method is simple and fast, and achieves high bandwidth efficiency for multiple QoS environment, it is suitable for future multimedia ATM networks.

1.4.4 A real-time MMPP parameter estimating method for calculating cell loss ratio

Since admission control is *preventive* control, we need to validate whether the admission control algorithm actually works properly. Since a role of CAC is to maintain the target CLR, we need to validate that the actual CLR is lower than the target value, which is extremely low, e.g., 1.0e-5 to 1.0e-8. Thus, it is hard to judge whether or not the target QoS level is maintained. We developed a real-time algorithm for estimating Markov modulated Poission process (MMPP) parameters to estimate the CLR [20, 21]. Once we can estimate the MMPP parameters, we can calculate the CLR by solving the MMPP/D/1/K queueing model [22, 23, 24, 25]. We consider that the aggregate cell stream fits to two state Markov Modulated Poisson Process (MMPP) models, and derive the parameters of the MMPP model. This method is based on a two-state cell stream model composed of overload and under-load states. Since the two state MMPP requires only four parameters, the proposed algorithm reduces the memory requirements to store the measurement data, which is suited to online processing. The developed algorithm employs a fictitious queue and a window to determine the load states: underload or overload. The performance of this method is investigated through simulation for multiplexing

of both homogeneous and heterogeneous on-off sources with a wide spectrum of traffic characteristics. We show that we can estimate the CLR properly by setting the window size to the minimum cell interval of the connections. We suggest that the proposed algorithm can be applied to admission control. Through computer simulation, we show that this admission control method achieves high bandwidth efficiency.

Chapter 2

Scalable ATM switch architecture based on Banyan network with cell bypass queueing discipline

Banyan networks are ones of multi-stage interconnection networks with self-routing capability which is suitable for ATM networks. It was already shown that the Banyan network composed of 2×2 switching elements with single buffer attains a maximum throughput of around 0.45. Even with an FIFO multiple buffer at each switching element, only a minor advantage can be obtained over the single-buffered Banyan network. It is because the head of line blocking occurs in the FIFO multiple buffer. In this chapter, we consider a non–FIFO queueing discipline at two input buffers of the 2×2 switching element, which we call a cell bypass queueing discipline to eliminate the influence of such head of line blocking. We present an exact analysis for the 2×2 switching element with input buffers based on this discipline. Furthermore, an approximate analysis is developed for the Banyan network composed of 2×2 switching elements with cell bypass queueing discipline. We investigate the effect of that discipline on performance through numerical results obtained by our analyses as well as simulation results. We also examine the nonuniform input traffic models. Our analysis shows that the cell bypass queueing discipline considerably improves performance of the Banyan network with respect to both throughput and cell blocking probability.

2.1 Introduction

In high speed communication networks, a large number of multi–stage interconnection networks such as Banyan networks have been proposed for an ATM switch owing to their self–routing capability and suitability for LSI implementations. However, blocking may occur at the internal switching elements in such Banyan networks; i.e., even if cells arriving at the different input trunks are destined for different output trunks, they may contend for the same internal link, and losers are blocked in the switching element. This results in performance degradation of the switch. Jenq [26] analyzed the Banyan network composed of 2×2 switching elements each of which has a single buffer on each input port, and showed that its maximum throughput is limited to around

0.45.

It is one of possible approaches for performance improvements to provide a multiple buffer, which can hold more than one cell, instead of a single buffer on each input port. Kim et al.[27, 28] analyzed the Banyan network with FIFO multiple buffers, and showed that it has a minor advantage over the one with single buffer. This is due to the effect of the head of line blocking in an FIFO input buffer. The head of line blocking refers to the following phenomenon. If cells at the head of buffers on both input ports are addressed to the same output port in a switching element, one of them is blocked. The other output port is thus left unused during the clock cycle even if there is a cell destined for that unused output port among the remaining cells being in the rest of the buffer.

The head of line blocking can be eliminated as follows; if the cell at the head of the input buffer loses an output contention, the following cells within the range of search (or scanning) are successively granted to contend for an output port until one of them wins the contention. This kind of a discipline was applied for $N \times N$ nonblocking packet switch in [29], and for each switching element in the Banyan network in [30]. Those papers gave simulation results for the maximum throughput and showed the improvement due to this discipline. Below, we will refer to the above discipline as a cell bypass queueing discipline. Note that it is called a *window selection policy* in [29] and a *bypass queueing discipline* in [30].

In this chapter, we first describe a detailed algorithm for this discipline in Sect. 2.2. Then, we will present an exact analysis for the 2×2 switching element with cell bypass queueing discipline in Sect. 2.3. Our analysis will provide the average delay time versus input rate and cell blocking probability at a finite buffer on an input port. These performance measures are obtained as a function of the range of scanning, which may be limited by hardware implementation. Furthermore, an approximate analysis is developed for the Banyan network composed of the 2×2 switching elements with cell bypass queueing discipline in Sect. 2.4. These analyses are able to be applied for nonuniform input traffic models. The effect of nonuniform input traffic will be examined for the 2×2 switching element in Sect. 2.3 and for the Banyan network in Sect. 2.4, respectively. Sect. 2.5 will conclude this chapter.

2.2 Buffered banyan networks with cell bypass queueing discipline

2.2.1 Structure of buffered Banyan network

The buffered Banyan network is a multistage network in which the switching elements with buffers are arranged in stages. Figure 2.1 shows a 3–stage (8×8) Banyan network. (More explanations for Fig. 2.1 will be given.) Each switching element is a crossbar switch with two input ports and two output ports, and has a multiple buffer on each input port to store one or more arriving cells apart from the "single-buffered" Banyan networks (e.g., [26]), which are composed of the 2×2 switching element with a single buffer on each input port. Each switching element is linked to the ones at its adjacent stages in such a way that a unique path can be established from every input trunk to every output trunk. A cell at stage n is automatically transferred to stage $n+1$ according to the n-th bit of its destination address of the output trunk if the input buffer at stage $n+1$ is not full, or if the buffer is full but the cell in the buffer can advance at the same clock cycle. If the corresponding bit is "0", the cell is passed to the upper output port while, in another case (the bit is "1"), it is passed to the lower output port.

The Banyan network is operated synchronously. In the first part of the clock cycle, control signals are

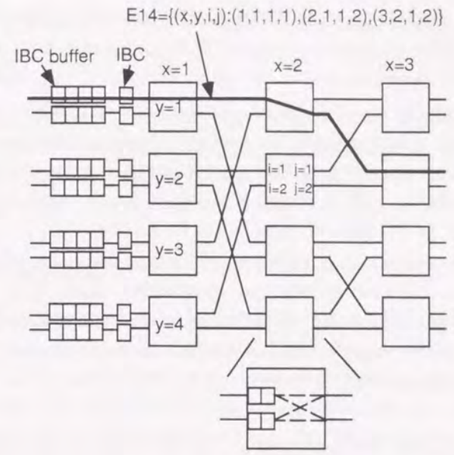


Figure 2.1: Sample 8 × 8 Banyan network.

passed across the network so that every switching element of the network can determine whether to send or hold its cells. Then, in the second part of the clock cycle, the cell is sent to the next stage in accordance with control signals. The whole process is repeated in every clock cycle.

As in [26] and [28], cells arriving on input trunks to the Banyan network are first stored in the buffer with input buffer controller (IBC) (see Fig. 2.1). At the beginning of the clock cycle, if the corresponding buffer of the switching element at the first stage is able to accept the cell, the IBC will move the cell into the buffer from the IBC buffer on an FIFO basis.

The following section will describe how to serve cells stored in two input port buffers of the switching element by the cell bypass queueing discipline.

2.2.2 Cell bypass queueing discipline for input buffer scheduling at switching element

Kim et al. [27, 28] analyzed the Banyan network with FIFO multiple buffer in which the cells in each input buffer are served in an FIFO manner. If cells at the head of two FIFO input buffers are destined for the same output port, one of these two cells will be randomly chosen to be passed to the next stage. Another cell is still stored in the buffer. The other output port is thus left unused during the clock cycle even if the cell destined for that output port exists in the rest of the buffers. This undesirable feature (called *head-of-line blocking*) degrades performance of the 2×2 switching element. Then, it leads to performance degradation of the entire Banyan networks.

In order to improve the performance of the switching element, a new (non–FIFO) discipline has been proposed in [29, 30]. A detailed algorithm of the discipline (refereed as a cell bypass queueing discipline) we consider in this chapter is as follows: if the cells at the head of both upper and lower input ports are destined

for the same output port (say output port 1), both buffers are scanned simultaneously from head to tail in order to find the cell destined for another output port 2. If the cell destined for an idle output port (output port 2 in this case) is found, it is served instead of the cell at the head of that buffer.

A *scanning range*, which specifies how far the cell destinations are checked from the head of the buffer, must be equal to the input buffer size to achieve the best performance. However, it may be limited to some smaller value than the buffer size by hardware limitation because each switching element should be operated at very high speed in the broadband ISDN. Thus, we will obtain the performance of the switch as a function of the scanning range which can be less than or equal to the buffer size.

More specifically, the cell bypass queueing discipline determines next transmitting cells as follows. Below, we assume that both of cells at the head of two input port buffers (input port buffers 1 and 2) are being destined for output port 1. With FIFO discipline, only one of two cells is allowed to be transmitted to the output port. On the contrary, we can consider the following four cases for determining cells to transmit in the case of the cell bypass queueing discipline.

Case 1: The cell destined for output port 2 is found only at input port buffer 1 within the scanning range.

The cell destined for output port 2 found at input port buffer 1 and the cell destined for output port 1 at the head of the input port buffer 2 are sent to the output ports.

Case 2: The cell destined for output port 2 is found only at input port buffer 2 within the scanning range.

The cell destined for output port 1 at the head of the input port buffer 1 and the cell destined for output port 2 found at input port buffer 2 are sent to the output ports.

Case 3: Both upper and lower input port buffers have cells destined for output port 2 within the scanning range.

The cell found first in the course of scanning both buffers is selected to send it to output port 2. As for the other input port, the cell at the head of it is sent to the output port 1.

Case 4: Any cell destined for output port 2 does not exist within the scanning range.

In this case, either of two cells at the head of input port buffers is randomly chosen and served in a clock cycle. Another unchosen cell is still held during at least one clock cycle.

When both of two cells at the head of input port buffers are destined for output port 2, cells to send are determined in a similar fashion. By this operation, we would expect that it is more possible to transmit two cells destined for distinct output ports in same clock cycle. Note that the sequence of the cells belonging to the same path cannot be broken even in using the above discipline. This fact is important in an ATM network environment.

Below, we first evaluate performance of a 2×2 switching element with cell bypass queueing discipline in the next section.

2.3 Performance of 2×2 switching element with cell bypass queueing discipline

In this section , we focus on performance of the 2×2 switching element. For this purpose, we first present an exact analysis of the switching element with cell bypass queueing discipline. Then, we will evaluate performance through the analytical results. Simulation results are also provided because our analysis has a limit such that it cannot be applied in the cases of buffers of large and a wide scanning range.

2.3.1 Modeling and analysis

Each 2×2 switching element consists of two input ports (input ports 1 and 2) two output ports (output ports 1 and 2). Each of two input ports has a finite capacity buffer of size L_i (i=1, 2). A scanning range is S_i (i=1, 2); $S_i \leq L_i$. Note that the case of S_i=1 corresponds to the classical Banyan networks with FIFO multiple buffers where destinations of only cells at the head of two input port buffers are examined. We assume that time is slotted with slot size being equal to a cell transmission time. Two of input lines synchronized . Cells arrive at the beginning of time slot, and then a scheduling to determine the next transmission cells to the output ports takes place. Its processing time is assumed to be equal zero.

We further assume that a cell arriving on input port i is destined for output port j with probability p_{ij} at each time slot, Namely, at input port 1, the cell arrives with probability $(p_{11}+p_{12})$ at each time slot, and the destination of the cell is the output port 1 [output port2] with probability $p_{11}/(p_{11}+p_{12})$ $[p_{12}/(p_{11}+p_{12})]$. At input port 2, same assumptions regarding cell arrivals are made.

Furthermore, we introduce a virtual probability r_j (j=1, 2) that the output port j is idle, and that it can accept the cell when the cell is being sent to output port j. With probability 1-r_j, the cell is not actually transmitted to output port j and still being held at the input port buffer at least until the next time slot. When we consider the whole Banyan network, this probability corresponds to the case where the input port buffer of the next stage, which is connected from the output port at the current stage, is available, or full but he cell at the head of that input buffer can advance at the same time slot. Note that r_j's are not required for the analysis of the switching element alone, but will be necessary when we analyze the entire Banyan network in the next section.

Since we have assumed that cells arrive according to a Bernoulli process, we can define the following discrete-time discrete-space Markov process. To describe this process, we introduce the following random variables. Let $d_i^j(t)$ $(1 \leq j \leq S_i)$ be a random variable indicating the output port number which, at time t, j-th. cell in input buffer i is destined for. Namely, when $d_i^j(t)$=1($d_i^j(t)$=2), j-th cell in the input port buffer i is destined for output port 1(output port 2). By $q_i(t)$, we define a random variable for the number of cells which are stored beyond the scanning range, the buffer state for each input port can be represented by only $d_i^j(t)$'s$(1 \leq j \leq k)$. Then the remaining $d_i^j(t)$'s $(k < j < S_i)$ and $q_i(t)$ are not required. We will use the vector $\boldsymbol{d}_i^j(t)$ for representing such a state. Namely,

$$\boldsymbol{d}_i^j(t) = (d_i^1(t), d_i^2(t), \cdots, d_i^k(t), -, \cdots, -), \qquad o \leq k \leq S_i, \tag{2.1}$$

where a symbol '$-$' means that the state is not applicable. In what follows, we will use the notation $N_j(\boldsymbol{d}_i^j(t))$ to represent the number of cells destined for output port j at input port buffer i for the given $\boldsymbol{d}_i^j$ for ease of presentation. Apparently, the sum of $N_1(\boldsymbol{d}_i^j(t))$ and $N_2(\boldsymbol{d}_i^j(t))$ equals k from its definition.

By these r.v.'s, we can completely specify the buffer state, $\boldsymbol{\Pi}_i(t)$, at input port buffer i at time t as:

$$\boldsymbol{\Pi}_i(t) = \begin{cases} (\boldsymbol{d}_i^j(t); -), & \text{if } k \leq S_i \\ (\boldsymbol{d}_i^j(t); q_i(t)), & \text{otherwise} \end{cases} \qquad i = 1, 2 \tag{2.2}$$

Then, we can represent the entire state of the switching element at time t by $\boldsymbol{\Pi}_i(t) = (\boldsymbol{\Pi}_1(t); \boldsymbol{\Pi}_2(t))$.

This process forms a Markov chain , and it is possible to obtain the transition probabilities $Prob[\boldsymbol{\Pi}_i(t+1)|\boldsymbol{\Pi}_i(t)]$. However , a description of such transition probabilities is a tedious work, and we omit it in the current chapter. By letting $t \to \infty$, we obtain the marginal steady state probability density for queue length distributions of two input port buffers: $P(\boldsymbol{\Pi}) = P(\boldsymbol{\Pi}_1; \boldsymbol{\Pi}_2)$. From now on, we will omit t in the above notations for representing that they are in the steady state.

2.3.2 Derivation of performance measures

Now, we derive the following performance measures for the 2×2 switching element.

Mean queue length

Below, we will derive the mean queue length at input port buffer 1. When the numbers of cells destined for output port 2 are m and $k - m$ (there exist k cells in total), respectively, the following two cases are considered:

Case1: All of k cells are within the scope of the scanning range, i.e., $k \leq \S_1$. Then, the probability that there exist m cells destined for output port 1 in the buffer can be derived by collecting the steady state probabilities, $P(\boldsymbol{\Pi})$, for all $\boldsymbol{\Pi}$ such that the number of cells destined output port 1, $N_1(\boldsymbol{d}_1^k)$, is equal to m.

Case 2: In the case where k ¿ S_1, we should take into account the destinations of $k - S_1$ cells which are located in the buffer beyond the scanning range because we only know the steady state probabilities for the total number of such cells (given by $q_1 = k - S_1$) by our state representations. When k' ($\leq \min(m, S_1)$) cells destined for output port 1 are within the scanning range, other $m - k'$ cells destined for output port 1 among $k - S_1$ cells is given by

$$p_1^{k,m,k'} = \begin{pmatrix} k - S_1 \\ m - k' \end{pmatrix} \left(\frac{p_{11}}{p_{11} + p_{12}}\right)^{m-k'} \left(\frac{p_{12}}{p_{11} + p_{12}}\right)^{(k-S_1)-(m-k')} \tag{2.3}$$

Now, by letting Q_{11} be the mean numbers of cells destined for output port 1, we have

$$\begin{aligned} Q_{11} = & \sum_{\forall \pi_2 \in \Pi_2} \{ \sum_{k=0}^{S_1} \sum_{m=0}^{k} \sum_{\forall \boldsymbol{d}_1^k s.t. N_1(\boldsymbol{d}_1^k)=m} m P(\boldsymbol{\Pi} = (\boldsymbol{d}_1^k; -); \pi_2) \\ & + \sum_{k=S_1+1}^{L_1} \sum_{m=0}^{k} \sum_{k'=0}^{\min(m,S_1)} \sum_{\forall \boldsymbol{d}_1^{S_1} s.t. N_1(\boldsymbol{d}_1^{S-1})=k'} m p_1^{k,m,k'} P(\boldsymbol{\Pi} = (\boldsymbol{d}_1^{S_1}; k - S_1); \pi_2) \} \end{aligned}$$

Q_{12} which is the mean number of cells destined for output port 2 at input port 1, is also obtained by the similar way. Furthermore, Q_{21} (Q_{22}), which is the mean number of cells destined for output port 1 (for output port 2) at input port 2, can be also obtained.

Blocking probabilities

When an input port buffer is full and any cell in the buffer can not advance the next stage, the cell arriving at the input port buffer will be blocked. Here, we will only consider the blocking probability, Pb_1, that the arriving cells at input port buffer 2, Pb_2, can be obtained by the same discussion.

To derive Pb_1, we consider the set of states such that the input port buffer 1 is full and the cells at the head of both upper and lower input port buffers are destined for output port 1, In this set of states, both upper and lower input buffers are scanned to find the cell destined for output port 2. This set can be classified into the following six subsets:

E_1: The subset of states such that the cells destined for output port 2 first appears at input port buffer 1. For example, the buffer states

$$\begin{aligned} \boldsymbol{\Pi}_1(t) &= (d_1^1 = 1, d_1^2 = 2, d_1^3 = 1, \cdots; q_1 = L_1 - S_1) \\ \boldsymbol{\Pi}_2(t) &= (d_2^1 = 1, d_2^2 = 2, d_2^3 = 1, \cdots; q_2 = L_2 - S_2) \end{aligned}$$

correspond to this case. The cell destined for output port 2 found at input port buffer 1 and the cell destined for output port 1 at the head of input port buffer 2 will be chosen to be sent to each output port. The probability that the chosen cell at input port buffer 1 fails to be sent to output port 2 is given by $1 - r_2$. Then, the next arriving cell at input port buffer 1 will be lost.

E_2: The subject of states such that the cell destined for output port 2 first appears at input port buffer 2. In this case, the cell at the head of the input port buffer 1 and the cell destined for output port 2 found at input port buffer 2 will be chosen to be sent to each output port. The probability that the chosen cell at the head of the input port buffer 1 fails to be transmitted to the output port 1 is given by $1 - r_1$.

E_3: The subset of states such that cells destined for output port 2 are located at the same distance from the head of buffers at both input port buffers 1 and 2. In this case, a cell to send to each output port is chosen from input port buffers 1 and 2 with equal probability 1/2. When the cell at the head of the input port buffer 1 is decided for the one to the output port 1, that cell cannot be sent to the output port 1 with probability $(1 - r_1)/2$. The probability of failing to transmit the cell is $(1 - r_2)/2$ when the cell destined for output port 2 is selected at input port buffer 1.

E_4: The subset of states such that the cell destined for output port 2 exists only at input port 1, and is never found at input port buffer 2 within the scanning range. For example, the following state illustrates this case:

$$\begin{aligned} \boldsymbol{\Pi}_1(t) &= (d_1^1 = 1, d_1^2 = 2, d_1^3 = 1, \cdots; q_1 = L_1 - S_1) \\ \boldsymbol{\Pi}_2(t) &= (d_2^1 = 1, d_2^2 = 1, d_2^3 = 1, \cdots; q_2 = L_2 - S_2) \end{aligned}$$

In this case, that cell will be sent to the output port 2 and fail it with probability $1 - r_2$.

E_5: The subset of states such that the cell destined for output port 2 exists only at input port buffer 2. In this case, cell will be chosen for being sent to the output port 2, but fail it with probability $1 - r_1$.

E_6: The subset of states such that the cell destined for output port 2 are not found neither at input port buffer 1 nor at input port buffer 2. The sample case is:

$$\begin{aligned} \boldsymbol{\Pi}_1(t) &= (d_1^1 = 1, d_1^2 = 1, \cdots, d_1^{S_1} = 1, \cdots; q_1 = L_1 - S_1) \\ \boldsymbol{\Pi}_2(t) &= (d_2^1 = 1, d_2^2 = 1, \cdots, d_2^{S_2} = 1, \cdots; q_2 = L_2 - S_2) \end{aligned}$$

In this case, the cell at states such that the input port buffer 1 will be chosen with probability 1/2 and fails to be sent to the output port with probability $1 - r_1$. If it is not chosen (this case occurs with probability 1/2), the arriving cell at input port buffer 1 will be blocked with probability 1.

Similarly, the set of states such that the input port buffer 1 is full and the cells at the head of both upper and lower input port buffers are destined for output port 2 can be classified into other six subsets. We denote those subsets as $E_7 \sim E_{12}$.

In summary, we obtain the blocking probability at input port 1, Pb_1 as

$$\begin{aligned} PB_1 &= (1 - r_2) \sum_{\forall \boldsymbol{\Pi} \in E_1 \cup E_4 \cup E_8 \cup E_{11}} P(\boldsymbol{\Pi}) + (1 - r_1) \sum_{\forall \boldsymbol{\Pi} \in E_2 \cup E_5 \cup E_7 \cup E_{10}} P(\boldsymbol{\Pi}) \\ &+ \frac{2 - r_1 - r_2}{2} \sum_{\forall \boldsymbol{\Pi} \in E_3 \cup E_9} P(\boldsymbol{\Pi}) + \frac{2 - r_1}{2} \sum_{\forall \boldsymbol{\Pi} \in E_6 \cup E_{12}} P(\boldsymbol{\Pi}) \end{aligned}$$

The blocking probability at input port 2, Pb2, can be derived in the same manner as in the above.

Mean delay time

By Little's Law,

$$D_{ij} = Q_{ij}/((1 - Pb_i)p_{ij}), \qquad i, j = 1, 2, \tag{2.4}$$

where D_{ij} is the mean delay time (including the cell transmission time) for cells from input port i to output port j.

2.3.3 Numerical discussions

In this section, we provide numerical results for performance of the 2×2 switching elememt with cell bypass queueing discipline. For numerical discussions, we set the scanning ranges for input port 1 and input port 2 to S, i.e., $S = S_1 = S_2$. Furthermore, both input ports have buffers of a same size L. The probability r_j is set to 1, i. e., the cells are never blocked at the output port when it wins the contention at the input port. The analysis presented above allows us to examine both uniform and nonuniform traffic patterns, which will be discussed in Sect. 2.3.3 and Sect. 2.3.3, respectively.

Uniform traffic pattern

For the uniform traffic case (i.e., $p = p_{11} = p_{12} = p_{21} = p_{22}$), we first give the mean delay time versus throughput observed at each output port in Fig. 2.2. The input port buffer size L is set to 4. Noting that the case of $S = 1$ corresponds to the classical FIFO discipline, we can see from Fig. 2.2 that the cell bypass queueing discipline offers a substantial improvement; e.g., the maximum throughput exceeds 0.9 with $S = 3$. The blocking probability is also improved by letting the scanning range be large as shown in Fig. 2.3.

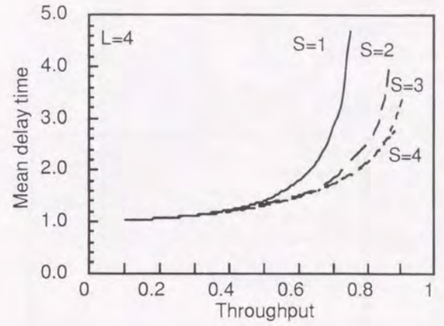


Figure 2.2: Throughput dependent on scanning range for the 2×2 switching element.

To examine the effect of a larger buffer size and a wider scanning range, we show the simulation results for the blocking probabilities for various values of scanning ranges S in Fig. 2.3. The arrival rate $p(= p_{11} = p_{12} = p_{21} = p_{22})$ is set to 0.45 i.e., a cell arrives on each input port with probability 0.9 at a time slot, and is destined for one of two output ports with equal probability. From the figure, we observe that the blocking probability can be improved with scanning range S of large size. Furthermore, Fig. 2.4 shows that the low blocking probability (e.g., 1e-6) can be obtained even with a rather short scanning range when the buffer size is fully large.

Nonuniform traffic pattern

Next, we examine the effect of nonuniform traffic pattern on the performance of the switching element. For this purpose, we consider two kinds of traffic; heavy traffic and light traffic at each input port. Then, we use the following three sets of performances for the traffic pattern: (p_{heavy}, p_{light}) = (0.85, 0.05), (0.72,0.18), (0.6,0.3) where p_{heavy} = p11 = p22 and p_{light} = p12 = p21 pheavy. The buffer size L set to 4.

In Fig. 2.5, we show the analytical results for the mean delay times for each set of parameters dependent on the scanning range S. In the figure, the notations, $D_{heavy}(= D_{12} = D_{21})$ and $D_{light}(= D_{12} = D_{21})$, are use d to represent the mean delay times for heavy traffics and for light traffic, respectively. Daverage is the average delay for both heavy traffic and light traffic, which is derived by $(p_{11}D_{11} + p_{12}D_{12})/(p_{11} + p_{12})$ $[= (p_{21}D_{21} + p_{22}D_{22})/(p_{21} + p_{22})]$. We also depict comparable results for uniform traffic case indicated by (0.45, 0.45). The corresponding blocking probabilities are shown in Fig. 2.6.

In Figs. 2.5 and 2.6, we can observe that the total average delay at the input port buffer ($D_{average}$) and the blocking probability can be improved by a wider scanning range even in the case of the nonuniform traffic pattern gives better performance with respect to the above performance measures. Note that the latter observation is a natural result because the contention between cells destined for the same output port is less likely to occur in the case of the nonuniform traffic pattern.

However, the mean delay time for the light traffic becomes larger as the degree of the unbalance between light traffic grows. To explain this, let us see such an extreme case as shown in Fig. 2.7. The numbers in the

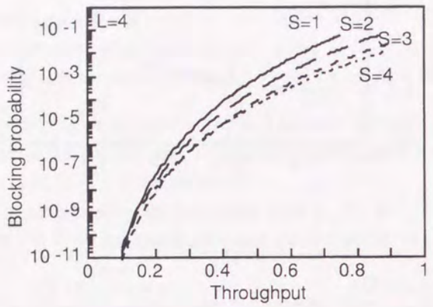


Figure 2.3: Cell blocking probability dependent on scanning range for the 2×2 switching element.

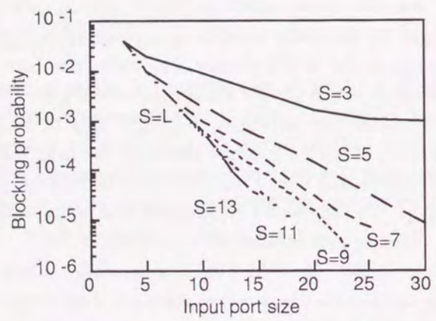


Figure 2.4: Cell blocking probability dependent on input port buffer length for the 2×2 switching element.

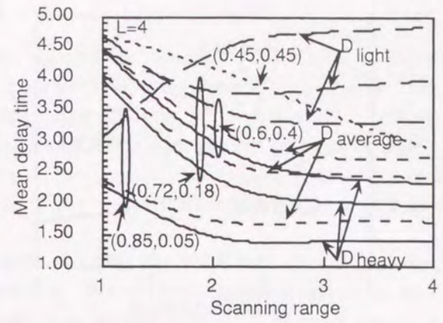


Figure 2.5: Effect of nonuniform traffic on mean delay times.

buffer us this figure show the destinations of the corresponding cells. Cells at the head of both of two input port buffers are destined for the same output port 2. Then, a scanning process to find the cell destined for another output port 1 is invoked. Such a cell is more likely to be found at input port buffer 1 under the traffic condition we now consider. The cell destined for output port 2 found in input port buffer 1 and the cell at the head of the input port buffer 2 are then transmitted while the cell destined for output port 2 at the head of input port buffer 1 is still held. Such a case occurs more and more as the scanning range becomes lager and/or the unbalance among the traffics becomes more extreme. This unfair service among traffic found in the cell bypass queueing discipline is an undesirable feature from a view point of "fairness " among traffic. However, there exists a tradeoff relationship between the scanning range and its obtained performance. It means that the unfairness can be restricted to some extent by a shorter scanning range. We will give more discussion on this infamies problem in Sect. 2.4.4.

Note that even with the case of $S = 1$, the delay for the light traffic is larger than the one for the heavy traffic as shown in Fig. 2.5. See Fig. 2.7 again, and suppose $p_{11} = p_{22}$ and $p_{12} = p_{21}$. The delay until the cell reaches the head of the buffer after its arrival must be identical among all traffic because the FIFO discipline is employed in the case of S=1. However, the cell destined for output port 2 at the head of input port buffer 2 sees the cell destined for output port 1 at the opposite side almost every time. Then, the contention for the output port is not likely to occur. On the other hand, since the case such that the cell destined for output port 2 at the head of input port buffer 1 finds the cell destined for output port 2 at the head of input port buffer 1can be transmitted to output port 2 with probability 0.5 at the time slot.

2.4 Performance of buffered Banyan networks with cell bypass queueing discipline

The Banyan network operates synchronously. So, we might model the buffer states of the entire Banyan network as a Markov process. Such an approach leads to us an exact analysis for the Banyan network. However, since an $N \times N$ Banyan network with $n(= \lceil \log_2 N \rceil)$ stages is composed of $n2^{n-1}$ switching elements, the number of states in an n-stage network becomes $k^{n2^{n-1}}$ where k represents the number of the buffer states in each

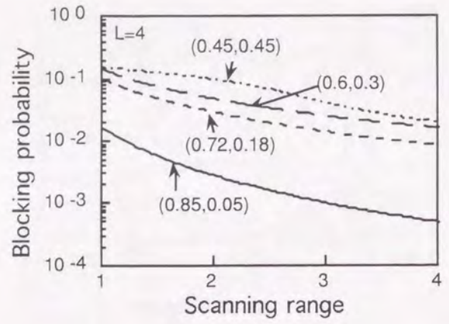


Figure 2.6: Effect of nonuniform traffic on cell blocking probability.

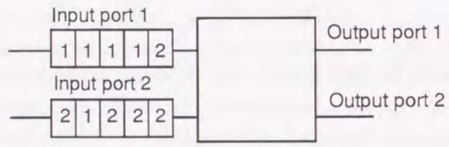


Figure 2.7: Buffer status under nonuniform traffic.

switching element. The number of states in the Markov chain grows exponentially with the number of stages n. Hence, an exact analysis using the Markov chain is intractable for a switch of large size. So, we need another approximate solution, for which a key idea is in the following. First, we solve each switching element and each IBC buffer in separately. Then, we modify the flow of the cells so that the total input rate at the first stage be equal to the total output rate at the last n-th stage. This implies that we should invoke the iterative solution method. This kind of an approach can be considered to be an extension of an analysis for tandem queueing networks with finite buffers (see, e.g., [31]).

Cells arriving in a time slot are first stored at the IBC buffers (Fig. 2.1). In the consecutive time slot, if the corresponding input buffer at the switching element of the first stage is able to accept the cell, the IBC will move the cell into that buffer from its own buffer. In the Banyan network, the cell will be routed as mentioned before.

In this section, we first introduce some notations, and analyze the IBC. Next, we will present an iterative solution method for analyzing the whole Banyan network. For this purpose, we will use an exact analysis for the switching element presented in the previous section. Lastly, we discuss the performance of the Banyan networks with cell bypass queueing discipline through numerical results computed by our analytical method and simulation.

2.4.1 Analytic model

In the $N \times N$ Banyan network, a path from the input trunk l to the output trunk m is determined uniquely. We use (x, y) to identify the upper y-th switching element at x-th stage (Fig. 2.1) where $1 \leq x \leq \lceil \log_2 N \rceil$ and $1 \leq y \leq N/2$. By letting i and j be the input port number and the output port number at each switching element, respectively, a path from input trunk l to output trunk m is expressed by E_{lm}; a set of (x, y, i, j). For example, the set $E_{14} = \{(x, y, i, j) : (x = 1, y = 1, i = 1, j = 1), (x = 2, y = 1, i = 1, j = 2), (x = 3, y = 2, i = 1, j = 2)\}$ indicates the path from input trunk 1 to output trunk 4 for the 8×8 Banyan network as shown in Fig. 2.1.

We also define the load matrix $\mathbf{\Lambda} = \{\lambda_{lm}\}$ in which each element, λ_{lm}, is an arrival rate for cells at input trunk l destined for output trunk m. We assume a Bernoulli arrival for cells at each input trunk. The cell arrival rate at switching element (x, y) is defined by $p_{ij}^{(x,y)}$ at input port i destined for output port j. Note that $p_{ij}^{(x,y)}$ can be calculated by collecting the appropriate flows of cells, λ_{lm}'s, if cell blocking does not occur in the internal switching elements of the Banyan network, i.e.,

$$p_{ij}^{(xy)} = \sum_{\forall l,m \text{ such that the set } E_{lm} \text{ contains element } (x,y,i,j)} \lambda_{lm} \tag{2.5}$$

Performance measures having been derived in the previous section for the switching element are also indicated by superscript (x, y) for the switching element (x, y), i.e., we will use $Pb_i^{(x,y)}$, $\gamma_{ij}^{(x,y)}$, and $D_{ij}^{(x,y)}$ in the below.

2.4.2 Analysis of the IBC buffer

IBC operates synchronously, so it can be modeled as a discrete-time Markov process. Let the buffer length of the l-th IBC buffer be L_{IBC_l}. Cells arrive at the l-th IBC buffer with probability p_{IBC_l} at each time slot, which is identical to the cell arrival rate at input trunk l given by $\sum_{m=1}^{m=N} \lambda_{lm}$.

A cell at the head of IBC buffer can advance the switching element at the first stage of the Banyan network with probability r_{IBC_l}. The probability, r_{IBC_l}, corresponds to the cell blocking probability at the input port buffer of the first stage in the Banyan networks. This probability will be obtained from the analysis of the switching element at the first stage as $Pb_i^{(1,y)}$ where the relationship $l = N/2 * (y - 1) + i$ holds.

We define the random variable $q_{\text{IBC}_l}(t)$ as the number of cells at the l-th IBC buffer at time t. In an IBC buffer, cells are assumed to be served in an FIFO manner, so transition probabilities $Prob[q_{\text{IBC}_l}(t+1)|q_{\text{IBC}_l}(t)]$ are easily obtained. The transition probability matrix is presented in Appendix B. Under the stationary condition (i.e., $t \to \infty$), we can derive the steady state probabilities $Prob[q_{\text{IBC}_l} = k]$ $(0 \leq k \leq L_{\text{IBC}_l})$ at the l-th IBC buffer. Using this probability density, we can compute the following performance measures for given r_{IBC_l} as follows:

Mean Queue Length

$$Q_{\text{IBC}_l} = \sum_{k=0}^{L_{\text{IBC}_l}} k \, Prob[q_{\text{IBC}_l} = k], \qquad 1 \leq l \leq N. \tag{2.6}$$

Blocking Probability

$$Pb_{\text{IBC}_l} = (1 - r_{\text{IBC}_l}) \, Prob[q_{\text{IBC}_l} = L_{\text{IBC}_l}], \qquad 1 \leq l \leq N. \tag{2.7}$$

Throughput

$$\gamma_{\mathrm{IBC}_l} = p_{\mathrm{IBC}_l}(1 - Pb_{\mathrm{IBC}_l}), \qquad 1 \le l \le N. \tag{2.8}$$

Mean Delay Time

$$w_{\mathrm{IBC}_l} = \frac{Q_{\mathrm{IBC}_l}}{\gamma_{\mathrm{IBC}_l}}, \qquad 1 \le l \le N. \tag{2.9}$$

2.4.3 Iterative solution method for the Banyan network

To analyze the entire buffered Banyan network with IBC's, the following iterative solution method is employed. The basic idea for this procedure has appeared in [28].

Step-1) Initialize the blocking probabilities, $Pb_j^{(x,y)}$'s, to be zero. The arrival rates at switching elements are then set to $p_{ij}^{(x,y)}$ given by eq.(2.5).

Step-2) Using the solution method described in Sect. 2.3 , analyze each switching element from the last stage to the first stage. For this purpose, we use the following quantities: the probability $r_j^{(x,y)}$, which equals the probability that the output port j is idle, is set to $1 - Pb_{\bar{j}}^{(x+1,\bar{y})}$. The probability $Pb_{\bar{j}}^{(x+1,\bar{y})}$ is the one having been derived as the blocking probability for input port $\bar{j}$ (which is connected to output port j of the current switching element (x, y)) of the switching element $(x + 1, \bar{y})$ at the next stage in this step. Note that $r_j^{(n,y)}$ for the switching elements at the last n-th stage is always zero.

Given that the input port i at the current switching element (x, y) is connected to the output port $\underline{i}$ of the previous switching element $(x - 1, \underline{y})$, the input rate $p_{ij}^{(x,y)}$ is set to the throughput of the output port $\underline{i}$ at the previous switching element $(x - 1, \underline{y})$ normalized by $1 - Pb_i^{(x,y)}$, i.e.,

$$p_{ij}^{(x,y)} = \frac{\{\gamma_{1,\underline{j}}^{(x-1,\underline{y})} + \gamma_{2,\underline{j}}^{(x-1,\underline{y})}\}}{1 - Pb_i^{(x,y)}}, \qquad 1 < x \le n. \tag{2.10}$$

At the switching elements of the first stage, the throughput of the IBC is used for $p_{ij}^{(1,y)}$.

By this analysis, the throughput $\gamma_{ij}^{(x,y)}$ and the blocking probability $Pb_i^{(x,y)}$ are derived, and will be used at Step-3 and Step-4.

Step-3) Analyze IBC buffers by the analytical method presented in Sect. 2.4.2. The probability, $r_{\mathrm{IBC}l}$, that the cell at the head of the IBC buffers can advance the first stage is given by the blocking probability at the first stage, $Pb_i^{(1,y)}$, which has been obtained in Step-1. The throughput of the IBC, which is identical to the input flow of cells entering the switching elements at the first stage, will be derived in this step.

Step-4) Do the analysis of the switching elements in a reverse order to the one in Step-1, i.e., from the first stage to the last stage. By this step, new quantities for blocking probability, $Pb_i^{(x,y)}$, and throughput, $\gamma_{ij}^{(x,y)}$, are obtained at each switching element (x, y) using the relationship $r_j^{(x,y)} = Pb_{\bar{j}}^{(x+1,\bar{y})}$ obtained at Step-2. For the input rate $p_{ij}^{(x,y)}$, the throughput at output port $\underline{i}$ of the previous switching element $(x - 1, \underline{y})$, which is derived at this step, can be utilized from eq.(2.10).

Step-5) Check the convergence of blocking probabilities at all switching elements and IBC buffers. Unless they converge (e.g., relative errors are smaller than 10^{-6}), go to Step-2.

After the above procedure is converged, the following performance measures can be obtained. The total cell delay time which consists of delays experienced at the IBC and switching elements are obtained by

$$w_{lm} = w_{\mathrm{IBC}_l} + \sum_{\forall (x,y,i,j) \in E_{lm}} D_{ij}^{(x,y)}. \tag{2.11}$$

By letting p_{Loss_l} be the cell loss probability at the input trunk l, we have

$$p_{\mathrm{Loss}_l} = Pb_{\mathrm{IBC}_l}. \tag{2.12}$$

As mentioned above, since switching elements and IBC's have been solved in separately, our results are approximate. We will check the accuracy of our procedure in Sect. 2.4.4.

2.4.4 Numerical discussions

For numerical discussions, we consider the 8×8 Banyan network. We assume that every IBC buffer has a same capacity $L_{\mathrm{IBC}}[= L_{\mathrm{IBC}_l}(1 \le l \le N)]$. We further assume that all switching elements have an identical structure, i.e., identical buffer sizes at two input ports in all the switching elements $[L = L_i^{(x,y)}, i = 1, 2$, for all elements $(x, y)]$, and the same scanning range $(S = S_i^{(x,y)}, i = 1, 2)$.

Uniform traffic pattern

We first consider the uniform traffic pattern in which every input trunk has the same rate for the arriving cells and they are destined for every output trunk with same probability $1/N$, i.e., $\lambda = \lambda_{lm}$ for all l and m.

In Fig. 2.8, the mean delay time as a function of obtained throughput are depicted for the case of $L_{\mathrm{IBC}} = 10$, $L = 4$, and $S = 1, 2, 3, 4$ using the analytical method presented in Sect. 2.4.3. Performance improvements obtained by the cell bypass queueing discipline are drastically even with a small size of the scanning range S. As reported in [28], with FIFO multiple buffer, the maximum throughput is rather low ($S = 1$ in our example). However, the figure shows that the cell bypass queueing discipline attains rather high throughput compared to the FIFO multiple buffer case. Fig. 2.9 confirms the above observation with respect to the cell loss probability at IBC buffers.

Since our analytical approach for the Banyan networks is approximation, the simulation results are also depicted in Fig. 2.8 to verify its accuracy. At light traffic load, differences between analytical results and the simulation results are very small. However, analytical results tend to underestimate the mean delay time compared to simulation results at heavy traffic loads. This tendency is also observed in [28] for the case of $S = 1$. The reason is that at heavy load, the modification of cell flows cannot be moderate since the network becomes saturated [28].

Nonuniform traffic pattern

Of many nonuniform traffic patterns, we study the particular nonuniform traffic pattern called SSSD (Single Source to Single Destination) in which all the cells arriving at a particular input trunk are destined for a

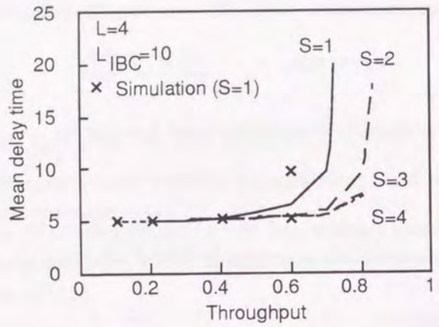


Figure 2.8: Mean delay times for the Banyan network under uniform traffic pattern.

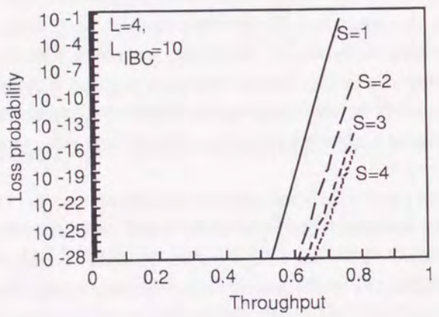


Figure 2.9: Cell loss probability for the Banyan network under uniform traffic pattern.

particular output trunk. Under SSSD traffic pattern, the Banyan network with FIFO buffers has already been studied in [28] where Kim et al. showed that the performance remarkably degrades in this case.

As described in Sect. 2.3.3, under nonuniform traffic, unfair treatment of traffic takes place at the switching element with cell bypass queueing discipline. Our main purpose of this section is to investigate the influence of such unfair treatment at each switching element on the entire performance of the Banyan network in addition to the effect of the cell bypass queueing discipline itself.

We add the heavy SSSD traffic as well as light traffic (which we call *background traffic* below) to the network. For the background traffic, we consider the uniform traffic pattern, and its arrival rate is denoted by λ. There exist four SSSD traffic which paths are $E_{11}, E_{35}, E_{67}, and E_{88}$.t Each arrival rate is given by λ_{SSSD}. The load matrix $\mathbf{\Lambda}$ is then given as:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda+\lambda_{\mathrm{SSSD}} & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda+\lambda_{\mathrm{SSSD}} & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda+\lambda_{\mathrm{SSSD}} & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda+\lambda_{\mathrm{SSSD}} \end{bmatrix} \tag{2.13}$$

The background traffic arriving at input trunk 1 destined for output trunk 8 are then considered to be given an unfair service at every stage as shown in Fig. 2.10 In the following numerical examples, we use the values of $\lambda = 0.0125$ and $\lambda_{\mathrm{SSSD}} = 0.8$.

By simulation, we obtain the mean delay times experienced in the network (including delays at the IBC buffer) as a function of the scanning range as illustrated in Fig. 2.11 The buffer size L of the switching element is set to 8. Note that we have used a simulation technique to obtain this figure because our analytical method has a limit so that it cannot be applied to the case of a wide scanning range as described before.

We choose representative mean delay times of interest. In the figure, w_{lm}'s represent the mean delay times for the background traffic arriving at input trunk l destined for output trunk m ($m = 1, 5, 7, 8$). Below, we call such background traffic T_{lm}. For comparison purpose, we also show the delays for the background traffic T_{33} which is not affected by the SSSD traffic at all stages.

The background traffic, T_{11}, T_{15}, T_{17}, and T_{18} shares the input link with the SSSD traffic at every stage. Then, the difference exists in how frequently they visit the switching element where the another input link has the SSSD traffic. As shown in the previous section, when two heavy SSSD traffic go straight through the different internal link at the switching element, the light background traffic crossing the switching element receives unfair service. This is a reason why the mean delay of background traffic T_{18} is larger than that of T_{17}. The difference is incurred at the switching element (3,4). The more important result observed in this figure is that the wider the scanning range is, the better performance is obtained for every background traffic even in the case of the traffic T_{18}. Figure 2.5 in Sect. 2.3.3 has shown that the mean delay of the light traffic crossing at the switching element becomes larger as the scanning range is wider. However, such a performance degradation is not observed for the light traffic T_{18} with a wider scanning range in Fig. 2.11. This is due to the fact that the performance improvement at other switching elements by introducing the cell bypass queueing discipline is dominant for the total delays. The background traffic T_{33} never encounters the SSSD traffic at every stage.

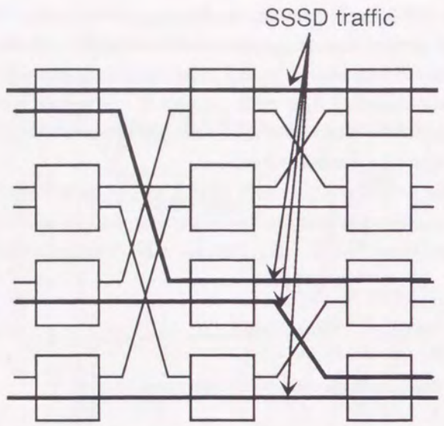


Figure 2.10: Nonuniform traffic pattern with background and SSSD traffic.

Then, its mean delay is almost same for various values of the scanning range. Figure 2.11 also shows that even with rather short scanning range, we can obtain low delays for every w_{lm} in the Banyan network.

Next, Figure 2.12 shows the sensitivity of the mean delay time to the buffer size at each switching element. We observe that the mean delay time is not sensitive to the buffer size when $S \geq 2$. It indicates that the mean delay time is much improved and the input port buffer of large size is not required once we have introduced the cell bypass queueing dicipline with short scanning range.

To obtain cell loss probability, we have used the analytic technique presented in Sect. 2.4.2. Figure 2.13 shows the cell loss probability at input trunk 1 dependent on the scanning range. For the input port buffer size, we have chosen $L = 4$. Note that it is difficult to obtain this kind of performance statistics by a simulation technique, which requires excessive computational cost for it.

2.5 Concluding remarks

In this chapter, we have investigated the effect of the cell bypass queueing discipline for the 2×2 switching elements in the Banyan network. For this purpose, we have first offered an exact analysis of the 2×2 switching element to derive the mean queue length at input port buffer, throughput, and cell blocking probabilities. Furthermore, an approximate analysis for the entire Banyan network was developed by employing the exact analysis for the switching elements. The simulation technique was also used to provide supplement results in the cases where our analytical method cannot be applied. Our results have shown that the Banyan network with cell bypass queueing discipline can offer a substantial improvement concerning the maximum throughput and the mean delay time over the Banyan network with FIFO discipline.

Our analyses can be also applicable to the cases of nonuniform input traffic models. The numerical results have shown that there is a significant difference between the mean delay time experienced by cells of the light traffic and those of heavy traffic in the switching element in some traffic situations. The differences between

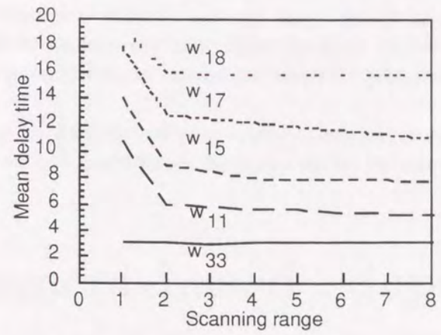


Figure 2.11: Mean delay times dependent on the scanning range for the Banyan network under nonuniform traffic.

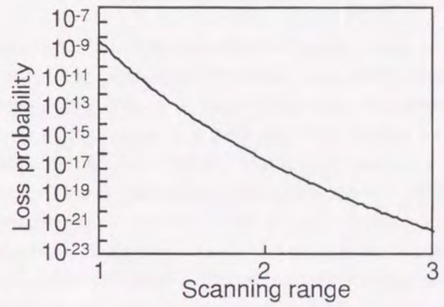


Figure 2.12: Mean delay times dependent on input port buffer length for the Banyan network under nonuniform traffic.

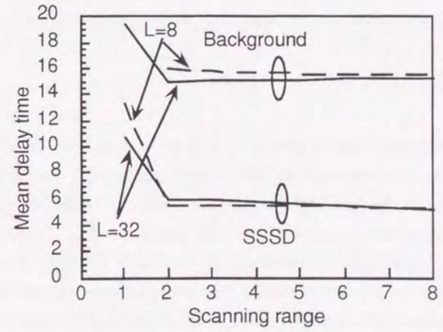


Figure 2.13: Cell loss probability at input trunk 1.

two kinds of traffic are larger under the cell bypass queueing discipline than under the FIFO discipline in the extremely unbalanced traffic situation. Such differences were also examined in the entire Banyan network, and it was observed that the mean delay time does not become larger with wider scanning range even in the extremely unbalanced case.

As for the analytic method, an approximate technique may be required to be applied to the cases of a wide scanning range and/or a large buffer size for the analysis of the switching element, which is left to be a future research.

Chapter 3

A measurement-based adaptive admission control

Efficiency of network resource can be improved by statistical multiplexing in ATM networks. If cell traffic characteristics of each connection could be obtained beforehand, we could admit maximum connections while satisfying the Quality of Service (QoS) objective. Since such traffic characteristics as an average rate and a mean burst length are difficult to anticipate, only peak rate will be used for connection admission control (CAC). The peak rate assignment strategy will, however, lead to inefficient network utilization for bursty traffic. In this chapter we present a measurement-based admission control method to tackle this problem. The design for a low-pass filter for ATM traffic measurement is presented and a framework for a bandwidth management based on it is proposed. We developed analytical model to investigate the relationship between the connection holding time and the distribution of the number of admitted connections. Simulation showed that the number of admitted connections is greater than that using conventional methods. Simulation showed that performance of the proposed method is not affected by long range dependent traffic. A simple implementation can be achieved by dividing the monitoring period into several bins and by expressing the smoothing coefficient of the low-pass filter by a power of two. Because this bandwidth management method does not assume any mathematical model and can be implemented in simple hardware, it is well suited for a practical ATM switching systems.

3.1 Introduction

In ATM networks, communications are performed in connection oriented manner. A connection is established between a source and a destination. The source sends cells at the peak rate when it has data to send and it doesn't send any cells when it has no data. Since multimedia traffic such as voice, video, and computer data are expected to be integrated into ATM networks, cell stream flowing over each connection is considered to be very bursty [32, 33, 34, 35, 36, 37, 38, 39]. Bandwidth management is performed at each link between adjacent nodes. For bursty traffic, the average rate of cell steam of connection is considered to be very low compared to the peak rate. If the peak rate assignment strategy were taken, the bandwidth efficiency would be very low. Efficient bandwidth management strategy needs to be studied to obtain high network resource utilization while maintaining high Quality of Service (QoS).

ATM networks use rate-based traffic control which uses the transmission rates for the individual connections for traffic control rather than sending credits back and forth between a user and the network. The user declares a traffic descriptor, such as the peak and average transmission rates at connection-setup time. The declared traffic descriptor is used to estimate the impact on network performance, and then a decision is made as to whether to admit the connection. This procedure is called connection admission control (CAC). However, there is disagreement about the reliability of the declared traffic descriptor [11, 40, 21, 20, 16, 41, 42, 43, 44, 45]. Even if the reliability of traffic descriptors for existing applications is good, new applications will emerge in future broadband networks. If we continue to use a traffic-descriptor-based approach, every time a new application is developed, an adequate traffic model will need to be constructed for it. A measurement-based approach is better because it does not need an accurate traffic characterization at connection-setup time.

To overcome this problem, a measurement-based Dynamic CAC methods have been proposed [11]. Dynamic CAC utilizes the measured cell arrival distribution [11]. The algorithm of the Dynamic CAC is summarized below.

- *Measurement of cell arrival distribution*:
  Let s and T denote the measurement window and measurement cycle, respectively. The empirical distribution $a(k;t)$ predicts that k cells will arrive in window s and be measured during the period[t-T,t]. The actual cell arrival distribution $\hat{a}(k;t)$ is evaluated using Eq (3.1) every T,

$$\begin{pmatrix} \hat{a}(0;t) \\ \cdots \\ \hat{a}(k;t) \\ \cdots \\ \hat{a}(M;t) \end{pmatrix} = \begin{pmatrix} \alpha a(0;t) + (1-\alpha)\hat{a}(0;t-T) \\ \cdots \\ \alpha a(k;t) + (1-\alpha)\hat{a}(k;t-T) \\ \cdots \\ \alpha a(M;t) + (1-\alpha)\hat{a}(M;t-T) \end{pmatrix}, \tag{3.1}$$

  where α denotes the smoothing coefficient.

- *Upperbound CLR*:
  If the new VC request whose peak rate is R [cells/window] arrives at time t, the upperbound $CLR(t)$ is determined as calculated below:

$$CLR(t) = \frac{\sum_{k=\max(K,R)}^{M+R} (k-K)\,\hat{a}(k-R;t)}{\sum_{k=R}^{M+R} k\,\hat{a}(k-R;t)}, \tag{3.2}$$

  where K denotes buffer size.

Dynamic CAC has been proven to achieve high efficiency for data communication [46]. Unfortunately, implementing the Dynamic CAC method requires very complicated circuits such as DSPs (Digital Signal Processor) and extra hardware logic [47]. This increases hardware complexity and system cost.

To overcome this hardware complexity problem, we propose a measurement-based CAC method that is considerably simpler than the Dynamic CAC method that requires estimates of the entire cell arrival distribution. In this chapter, we describe our design for a measurement-based bandwidth management method. In designing this method, we had three goals. The first goal was to design a method that would be efficient and would not degrade performance. A trade-off exists between bandwidth efficiency and performance: as the bandwidth efficiency increases, the network performance degrades. The target network performance can be

maintained by regulating the number of connections. Since maximum cell transfer delay is determined by the buffer size and the link capacity, we didn't consider the cell transfer delay. Our study assumed that the cell loss ratio (CLR) and the underload period should be considered as network performance indecies. Adequate measures for estimating these indecies should be selected.

The second goal was to minimize the dependency on a specific mathematical model for the input traffic. Several mathematical traffic models are in use, such as the Poission process, the interrupted Poisson process (IPP), the Markov-modulated Poisson process (MMPP), and the autoregressive (AR) model. However, all connections cannot necessarily be classified into one of these models. In addition, analyzing a queueing model loaded using one of these models is too complicated to be performed within the short period required for bandwidth management, say a few to tens of milliseconds.

The third goal was to minimize complexity because lengthy computation increases the set-up delay. In addition, expensive extra hardware and software should not be needed. To reduce the switching-node cost, the method should be simple enough to implement in hardware and/or software.

The rest of this chapter is organized as follows. In Section 3.2, we examine the effect of buffering on the CLR and demonstrate that an instantaneous rate, i.e., the total cell rate from active connections, is crucial to bandwidth management. In Section 3.3, we present a method for designing a low-pass filter (LPF) that can obtain the instantaneous rate from crude measurements (time series data for the number of cells in a slot). We present a method for designing both a moving average type and a recursive type LPF and demonstrate that the recursive type one performs stably, regardless of the connection's peak rate. In Section 3.4, we present a bandwidth management framework based on the low-pass filtered instantaneous rate. The maximum instantaneous rate observed over a monitoring period is used for the framework. We analyze the distribution of underload period and demonstrate how the monitoring period should be dimensioned. In Section 3.5, we evaluate the performance of our proposed bandwidth management framework, taking into account connection arrival and departure. We analyze a distribution of admitted connections and discuss the relationship between the average number of connections and the mean holding time. Using computer simulation results, we compare the performance of our proposed method with that of a conventional CAC [11]. In addition, we investigate the impact of long range dependent (LRD) traffic on the performance of our proposed method. In Section 3.6, we address implementation issues and present an example implementation. By dividing the monitoring period into several bins and expressing the smoothing coefficient of the recursive LPF by a power of two, we can implement our bandwidth management method by using a simple hardware. We also compare the computational complexity of our proposed method with that of conventional CAC. We show that the computational complexity of our proposed method is always lower irrespective of the size of the switching fabric. In Section 3.7, we summarize our key points and mention future work.

3.2 Measurement item –Instantaneous rate–

3.2.1 Effect of buffering on CLR

ATM traffic is characterized by three levels of fluctuation: cell, burst, and connection level [48]. The burst-level fluctuations in particular must be considered when designing admission controls [49]. Figure 3.1 shows the relationship between the CLR and buffer size, where N connections are multiplexed into a link. Each connection is modeled by on-off sources whose on and off periods are exponentially distributed. Cells are sent

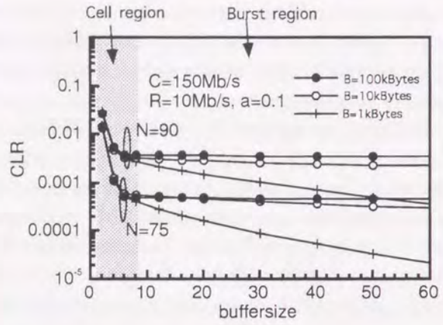


Figure 3.1: Relationship between CLR and buffer size.

at peak rate of $R = 10$ Mbit/sec during an on period; mean burst size $B = 1$, 10, or 100 Kbytes; burstiness factor $a = 10$.

In the region where the buffer size is less than around ten cells, the CLR decreases sharply as the buffer size increases. This slope is determined by how often cells from different connections arrive simultaneously in a certain cell slot. It is approximated by an $M/D/1/K$ model with the same average offered load [50, 51]. This region is called the cell-level region. In it, we can reduce the CLR dramatically by enlarging the buffer.

In the region where the buffer size is larger than around ten cells, the slope becomes gentle. This region is called the burst-level region. In it, the buffer should be sufficiently large to absorb the excess bursts when overload conditions arise. As shown in Fig. 3.1, the reduction in CLR due to enlarging the buffer depends on the mean burst length. Thus, in the burst-level region, we must consider the burst length distribution in order to keep the CLR lower than the target value. However, estimating the burst length distribution at connection setup time is difficult. Moreover, recent research shows that actual LAN traffic exhibits heavy-tailed distribution, which cannot be absorbed using an ATM buffer of a feasible size [39]. Accordingly, we cannot rely on buffering the excess bursts to keep the CLR lower than the target.

In Fig. 3.1, the CLR at the knee point corresponds to the virtual cell loss probability in buffer-less fluid-flow model [52, 51]. Note that the virtual cell loss probability does not depend on the burst length distribution. Rather, it depends on the number of connections (and their peak and average rates). Roughly speaking, the CLR at the knee points is determined by how often the simultaneously active connections exceed the link capacity. Consequently, capturing burst-level fluctuations is important for keeping the CLR below the target. The significance of burst-level fluctuations was also shown by spectral analysis of queueing systems [53, 54]. Therefore, we focused our attention on burst-level fluctuations.

3.2.2 Instantaneous rate

We use the concept of the *instantaneous rate* to capture the burst-level fluctuations. Burst-level fluctuations are characterized by the number of active connections, as illustrated in Fig. 3.2. The instantaneous rate is defined as

$$\lambda(t) = \sum R_i/C, \tag{3.3}$$

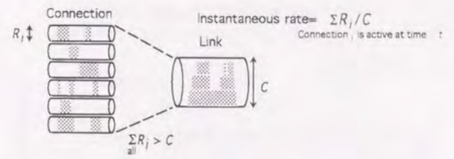


Figure 3.2: Conceptual view of definition of instantaneous rate.

where R_i denotes the peak rate of active connection i. The number of active connections changes with time, and so does the instantaneous rate. The instantaneous rate has also been recognized as an important measurement item by other researchers independently [41, 44].

3.3 Design of low-pass filter

3.3.1 Frequency analysis of cell stream

The instantaneous rate is the sum of the peak rate of the active connections. Analysis of a cell stream over a single connection in a frequency domain shows that eliminating frequency components higher than the connection's peak rate enables the instantaneous rate to be determined.

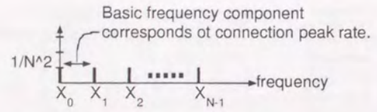
Assume that a link capacity is 150 Mbit/sec, a connection's peak rate is 10 Mbit/sec, and that a cell stream over the connection is modeled by an on-off source. The connection alternates between on- and off- states, sending cells at the peak rate during an on-state, and not sending cells during an off-state.

When the connection is in an on-state, the cell arrival pattern over the connection is periodic, with a cycle of 15 cell slots. By calculating the discrete Fourier transform (DFT) of the number of cells per cell slot, we obtain $X_i = 1/N^2$ for $i = 0 \ldots N-1$, where the basic frequency corresponds to 10 Mbit/sec, $N = 15$, and X_i denotes an i-th harmonic component. This result implies that a set of harmonic frequencies whose basic frequency is the connection peak rate constitutes a cell stream over a single connection (see Fig. 3.3(a)). The relationship between direct current (DC) component X_0 and connection's peak rate r (normalized by the link capacity) is $r = \sqrt{X_0}$ [53].

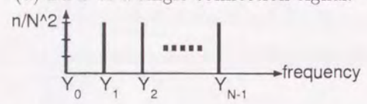
When n connections now are simultaneously in an on-state, the DFT of the multiplexed connections is $Y_i = nX_i$ for $i = 0 \ldots N-1$ due to the linearity of the DFT (see Fig. 3.3(b)). The sum of the peak rates of the connections in an on-state is $\sqrt{Y_0} = nX_0$. Consequently by eliminating Y_i, $i = 1 \ldots N-1$, we obtain Y_0, which is equal to the sum of the peak rates of connections in the on-state, i.e., the instantaneous rate. Therefore,to determine the instantaneous rate, those frequency components higher than the connection's peak rate should be removed.

3.3.2 Sliding window type low-pass filter

A sliding window can be used to police individual connection traffic [55], where the rate is given by the number of cells X divided by the window size T cell slots. This LPF is regarded as a moving average in a digital signal processing theory [56]. The instantaneous rate obtained using a moving average whose sample size is T is given



(a) DFT of a single connection signal.



(b) DFT of multiplexed connections signal.

Figure 3.3: Representation of the DFT of signal over (a) a single and (b) multiplexed connections in the frequency domain

.

by

$$\lambda(t) = \frac{1}{T}\sum_{k=0}^{T-1} n(t-k). \tag{3.4}$$

The power spectral function of Eq. (3.4) is given by Eq. (3.5) (for details see appendix B).

$$S(\omega) = \frac{1}{T^2}\sum_{k=0}^{T-1}(\cos^2(k\omega T_s) + \sin^2(k\omega T_s)), \tag{3.5}$$

where $\omega = 2\pi f$ and f denotes the signal frequency and T_s is the cell slot time over the link.

Figure 3.4 shows the power spectral function of a sliding window LPF with a window size of 15 cells. The link capacity is 150 Mbit/sec. We observe that this LPF filters out the frequency component of the 10 Mbit/sec signal and its harmonics, i.e., $10n$ Mbit/sec. When we apply the sliding window LPF to a purely periodic signal with a peak rate of 10 Mbit/sec, there is no problem. However, a problem arises when it is applied to a signal composed of various frequency components. Suppose that there are two connections whose peak rates are 9 and 10 Mbit/sec. If we use a sliding window LPF tuned to the 10 Mbit/sec signal, the harmonic frequencies of the 9 Mbit/sec connection appear as noise. Note that the variance in the power spectrum function in Fig. 3.4 is steep around a frequency of $10n$ Mbit/sec. As a result, the noise due to the 9 Mbit/sec connection is large.

3.3.3 Recursive low pass filter

Another type filter is the recursive low-pass filter (LPF), whose smoothing coefficient is α in Eq. (3.6).

$$\lambda(t) = \alpha n(t) + (1-\alpha)\lambda(t-\Delta), \qquad 0 \le \alpha \le 1, \tag{3.6}$$

Smoothing coefficient α controls the cut-off frequency of the LPF. As it approaches 1, the cut-off frequency increases. The power spectral function of Eq. (3.6) is given by Eq. (3.7) (for details see appendix B).

$$S(\omega) = \frac{\alpha^2}{1 + (1-\alpha)^2 - 2(1-\alpha)\cos(\omega T_s)}. \tag{3.7}$$

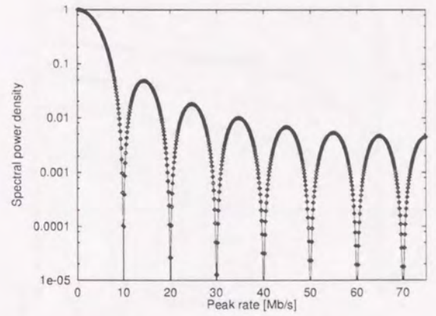


Figure 3.4: Power spectral function of the sliding window LPF. Link capacity is 150 Mb/s, and window size is 15 cells.

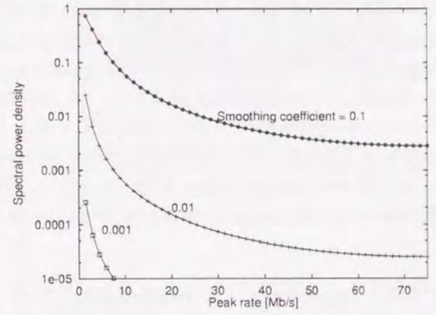


Figure 3.5: Power spectral density function of recursive low-pass filter. Link capacity is 150 Mb/s.

Figure 3.5 shows the power spectral density function of a recursive LPF. We observe that the power spectral density decreases as the connection's peak rate increases. To eliminate those frequency components higher than the connection's peak rate, we set α so that the corresponding $S(\omega)$ will be negligibly small, i.e., ϵ. Such an α is given by Eq. (3.8).

$$\alpha = \frac{-2(1-K) + \sqrt{4(1-K)^2 + 8(\epsilon^{-1}-1)(1-K)}}{2(\epsilon^{-1}-1)}, \tag{3.8}$$

where $K = \cos(\omega T_s)$. Figure 3.6 shows α as a function of the connection's peak rate. It shows that we should set α to 4.156e-3 when connections having a peak rate of 10Mbit/sec are multiplexed onto a link having a capacity of 149.76 Mbit/sec.

We conducted computer simulation to demonstrate the effect of the recursive LPF. We obtained time series data of the crude measurement data (i.e., the number of cells counted per unit time), an actual instantaneous rate, and two rates calculated using different smoothing coefficients. We assumed that the link capacity (C) = 149.76 Mbit/sec, the number of connections (N) = 50, the mean burst length of connections (B) = 100k

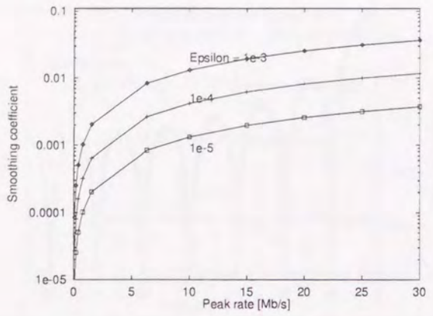


Figure 3.6: Relationship between α and a connection's peak rate. Link capacity is 150 Mb/s.

Bytes, the peak rate of connections (R) = 10 Mbit/sec, and the burstiness factor (a) = 0.1.

The crude measurement data and actual instantaneous rate are shown in Figs. 3.7(a) and 3.7(b), respectively. The crude measurement data varies greatly from the actual rate: the crude measurement data fluctuates strongly and the values are widely distributed because they reflect the cell-level fluctuations. Figs. 3.8 (a) and 3.8 (b) show the sample paths of instantaneous rates observed using LPFs with different α values. A large α (=1.0e-1) did not eliminate the cell-level fluctuations (Figs. 3.8(a)), while an adequate α (=4.156e-3) yielded an instantaneous rate close to the actual one (compare Figs. 3.7(b) to 3.8(b)). Thus, if smoothing coefficient α is set according to Eq. (3.8), we can obtain an instantaneous rate very close to the actual one.

3.4 Framework for the bandwidth management

3.4.1 Maximum instantaneous rate and residual bandwidth

Our proposed bandwidth management strategy, which uses measured instantaneous rate, is illustrated in Fig. 3.9. The cells arriving during a cell-transmission time over a link are counted and the cell count is then converted into the instantaneous rate by an LPF.

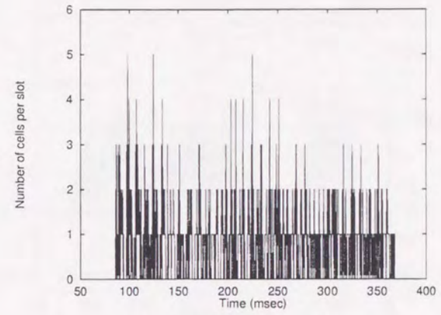
The instantaneous rate is monitored and the maximum instantaneous rate is defined as the maximum value observed during the monitoring period. The maximum instantaneous rate $\Lambda(t)$ is a function of time:

$$\Lambda(t) = \max_{t' \in (t-T_m, t]} \lambda(t'), \tag{3.9}$$

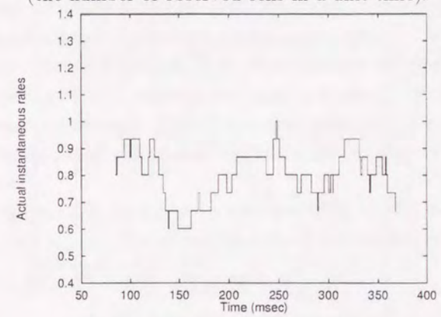
where T_m denotes the monitoring period. The residual bandwidth is derived from the maximum instantaneous rate as $C - \Lambda(t)$.

3.4.2 Admission decision

When a new connection setup request is received, the requested bandwidth R is compared to the current residual bandwidth. If the requested bandwidth is smaller than the residual bandwidth, the connection is

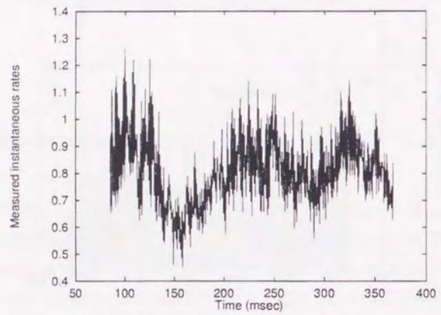


(a) Crude measurement
(the number of observed cells in a unit time).

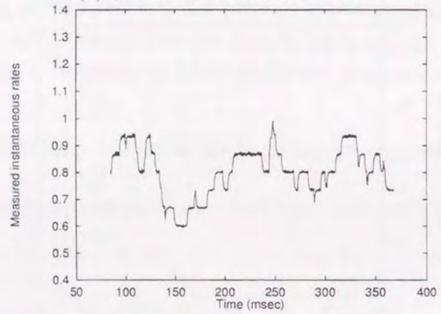


(b) Actual instantaneous rate.

Figure 3.7: Sample paths of (a) crude measurement and (b) actual instantaneous rate from computer simulation. Link capacity is 149.76 Mb/s, number of connections is 50, connection peak rate is 10 Mb/s, burstiness factor is 10, burst size is 100 KBytes.



(a) Measurement with α=1.0e-1.



(b) Measurement with α=4.156e-3.

Figure 3.8: Sample paths of filtered instantaneous rates from computer simulation. Link capacity is 149.76 Mb/s, number of connections is 50, connection peak rate is 10 Mb/s, burstiness factor is 10, burst size is 100 KBytes.

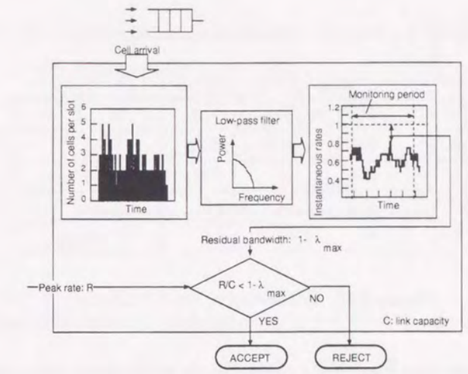


Figure 3.9: Concept of proposed bandwidth management strategy.

accepted; otherwise, it is rejected. Thus the admission criteria for acceptance is

$$R < C - \Lambda(t), \tag{3.10}$$

for a new connection-setup request whose peak rate is R, arriving at a link whose capacity is C.

The monitoring period is determined by network management policy. The longer the monitoring period, the more conservativeness is our proposed method. A short monitoring period may lead to frequent congestion. When this occurs, reactive control such as connection rerouting or dynamic VP capacity control should be taken. Extra operational cost increases accordingly. During a period when such reactive controls are invoked, QoS will be degraded. The monitoring period should thus be determined so that the frequency of QoS degradation remains at an acceptable level.

The T_m is related to the target CLR. To achieve a very low CLR, the T_m needs to be very long. To reduce the T_m, we introduce target load λ_{target}. The admission criteria is modified accordingly:

$$R/C < \lambda_{target} - \lambda_{max}(t). \tag{3.11}$$

The effect of λ_{target} and the relationship between the T_m and the target CLR will be discussed in more detail in Section 3.4.4.

3.4.3 Conservative modification

After a new connection is admitted, characteristics of the aggregate cell flow may change substantially. If the effect of the new connection is not considered, congestion is likely to occur. To avoid this, we take into account the effect of the new connections by adding to the measurement the declared peak rate of the connection.

If the connection is accepted, the observed instantaneous rate increases by an amount equal to the accepted connection's peak rate (see Fig. 3.10):

$$\lambda(t') = \lambda(t') + R, \qquad \text{for } t' \in [t - T_m, t]. \tag{3.12}$$

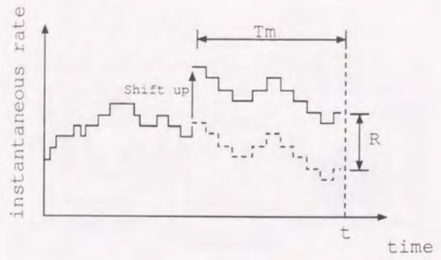


Figure 3.10: Conservative modification.

Even if the newly admitted connection sends cells continuously at the peak rate, the instantaneous rates will not statistically exceed the link capacity for the monitoring period, provided that the admission criteria given in Eq. (3.10) are satisfied.

3.4.4 Monitoring period

As discussed in Section 3.4.2, the monitoring period needs to be dimensioned so that the actual CLR is lower than the target CLR. The T_m depends on the target CLR and the traffic characteristics of the multiplexed connections. We will now examine the extent to which the T_m is affected by the traffic characteristics. We derived the T_m by approximating the 99% cumulative value of the distribution of the underload periods. An underload period is defined as one in which the actual instantaneous rate does not exceed the target load.

Analysis of distribution of underload period

To derive the distribution of the underload periods in which N on-off sources are multiplexed, we assume that the durations of the on- and off- periods are distributed exponentially with mean of ω^{-1} and β^{-1}, respectively. The number of connections in an on-state at time t follows a birth-death process denoted by $N_b(t)$.

Let T be a random variable for the duration of an underload period. The T is the absorption time of a transient Markov process whose state space is $\{N_b(t) : N_b(t) = 0, \ldots, N_u\}$. An absorbing state is $N_u + 1$, where N_u $(= \lfloor \lambda_{target} C/R \rfloor)$ denotes the border between an underload and overload state. (If more than N_u sources are in an on-state, that state is regarded as an underload state.)

The probability density function for the absorption time is given by

$$f(t) = \boldsymbol{\delta} \exp(\boldsymbol{Q}t)\boldsymbol{Q}^o, \tag{3.13}$$

where

$$\boldsymbol{\delta} = (0, \ldots, 0, 1), \tag{3.14}$$

$$\boldsymbol{Q}^o = (0, \ldots, 0, (N - N_u + 1)\beta)^T, \tag{3.15}$$

Table 3.1: Maximum number of connections supporting a CLR of 1.0e-6.

R (Mbit/sec)	a=10	a=20	a=100	a=200
1.5	670	1320	6527	13038
3	275	538	2640	5268
6	102	195	944	1880
10	44	82	388	770
20	11	18	78	153

and $\boldsymbol{Q}$ is an infinitesimal generator whose (i, j)-elements, Q_{ij}, are given in the following expression.

$$Q_{ij} = \begin{cases} i\omega, & i = j + 1 \\ -i\omega - (N - i)\beta, & i = j \\ (N - i)\beta, & i = j - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

The first and second moments of T are given by the following formulae [57, p. 45]:

$$\mathrm{E}[T] = \boldsymbol{\delta}(-\boldsymbol{Q}^{-1})^2\boldsymbol{Q}^o \tag{3.17}$$

$$\mathrm{E}[T^2] = 2\boldsymbol{\delta}(-\boldsymbol{Q}^{-1})^3\boldsymbol{Q}^o \tag{3.18}$$

Standard deviation σ is calculated from the first and second moments. By applying Chebyshev's inequality [58, p. 388], we can determine the 99% point of the distribution of T:

$$Prob(|T - \mathrm{E}[T]| \geq k\sigma) \leq \frac{1}{k^2}. \tag{3.19}$$

Numerical example

Here we evaluate the effect of the traffic characteristics on the T_m, assuming that the target CLR is 1.0e-6. Table 3.1 shows the maximum number of connections for traffic with different characteristics over a link with a capacity of 149.76 Mbit/sec. The number of connections is calculated using the virtual cell loss probability [52].

We approximated the 99% cumulative value of the underload periods under the conditions listed in Table 3.1. The effect of the peak rate, the burstiness factor, and the mean burst length on the T_m are plotted in Figs. 3.11, 3.12, and 3.13, respectively.

We can observe the following from these figures.

- The monitoring period can be shortened by introducing λ_{target}. A λ_{target} of 0.8 reduces the T_m to only 1/10th to 1/100th times that required for the normal condition, i.e., $\lambda_{target} = 1.0$.
- The monitoring period depends on the mean burst length, but is little affected by the burstiness factor or peak rate. This means the maximum burst length should be limited or the T_m should be adaptively modified by using a feedback mechanism. A T_m of 10 is sufficiently large for the entire range of both burstiness and peak rate (with $\lambda_{target} = 0.8$).

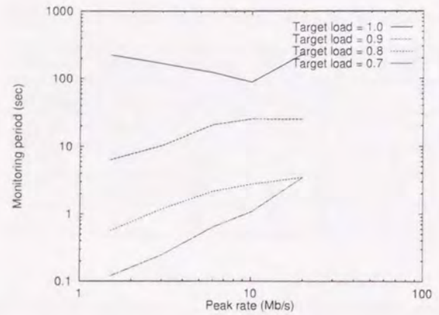


Figure 3.11: Impact of peak rate on T_m. Link capacity is 149.76 Mb/s, and CLR is 1.0e-6.

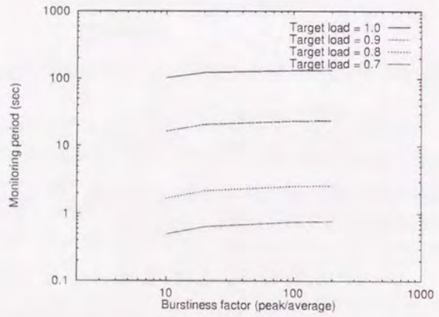


Figure 3.12: Impact of burstiness on T_m. Link capacity is 149.76 Mb/s, and CLR is 1.0e-6.

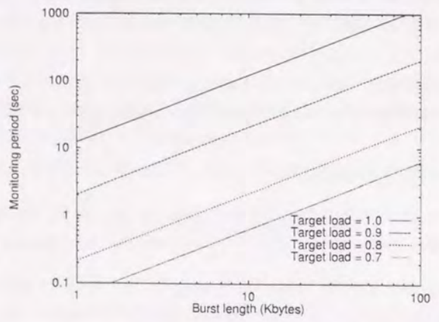


Figure 3.13: Impact of mean burst length on T_m. Link capacity is 149.76 Mb/s, and CLR is 1.0e-6.

3.5 Performance evaluation of bandwidth management method

To evaluate the performance of the proposed bandwidth management method, we investigated three points: (1) how many connections are admitted by our method, (2) how does it compare with conventional CAC methods, and (3) what is the impact of long range dependent (LRD) traffic. We developed an analytical model to investigate (1), and conducted computer simulation to investigate (2) and (3).

3.5.1 Analysis of number of admitted connections

An analytical model we developed to determine the number of connections admitted by our method assumes that the connection arrival rate and holding time distributions are given. We used it to calculate the relationship between the average number of connections and the mean holding time for various monitoring periods. As will be shown later, the average number of connections becomes saturated as the mean holding time increases.

Modeling

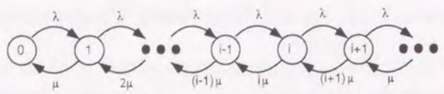
We assume that the arrival process of connections is Poisson process with parameter of λ [sec^{-1}] and the connection holding time is exponential distribution with mean of μ^{-1} [sec]. Once the connection is setup, packets are generated over the connection. Packet arrival process over each connection is modeled by an on-off model. When the source sends a packet, it is said to be in the on-state. When the source does not send a packet, it is said to be in the off-state. During the on-period, the packet is segmented into cells and the cells are sent out at the peak rate R. The packet size is exponential distribution with mean of β^{-1} Bytes. The inter-arrival time of the packet is exponential distribution with mean of α^{-1}.

Let $N(t)$ be the number of connections at time t. Without admission controls, $N(t)$ follows the birth-and-death process shown in Fig. 3.14 (a). With admission control, $N(t)$ follows the birth-and-death process shown in 3.14(b). In Fig. 3.14(b), γ_k is the factor for multiplying the upward transition from the state $N(t) = k$. It depends on the number of connections and admission control algorithm. With measurement-based admission controls, γ_k is a random variable. With our proposed method, if the residual bandwidth is larger than R, any connection requesting a bandwidth less than R is accepted. We thus redefine a underload period as one that starts when the instantaneous rate drops below the link capacity minus the requested bandwidth; it ends when the instantaneous rate rises above the link capacity minus the requested bandwidth. The probability of the underload period exceeding the T_m determines γ_k. Letting $F_k^C(t)$ be a complementary probability distribution function of an underload period when k connections are in progress, we have

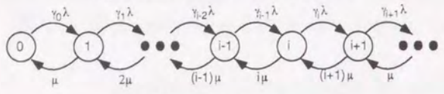
$$\gamma_k = F_k^C(T_m) = Prob\{\text{underload period} > T_m\}. \tag{3.20}$$

Distribution of underload periods

Suppose that there are N connections are in progress. The number of active connections is a birth-and-death process. We split the state space of the birth-and-death process into two subsets: underload and overload. The state is an underload one if the instantaneous rate is less than the link capacity minus the requested bandwidth. Otherwise it is an overload state. We neglect a trivial case where the instantaneous rate never exceeds the link capacity minus the requested bandwidth, i.e., $N \leq I$, where $I = \lfloor \frac{C-R}{R} \rfloor$.



(a)Without admission control.



(b)With admission control.

Figure 3.14: State transition diagram for $N(t)$.

Consider a Markov process for the underload state $\{0, \ldots, I+1\}$. The probability that there is no overload period during the monitoring period can be regarded as the probability that the first passage time of this Markov process is longer than T_m. The first passage time is defined as the time for the Markov process starting at the initial state to reach $I+1$. The distribution of the first passage time is a phase-type distribution, which is defined by the initial probability vector and the infinitesimal generator [57]. The infinitesimal generator of the Markov process is given by

$$\begin{pmatrix} \boldsymbol{Q}_u & \boldsymbol{Q}_u^o \\ \mathbf{o} & 0 \end{pmatrix}, \tag{3.21}$$

where

$$\boldsymbol{Q}_u = \begin{pmatrix} c_0 & b_0 & 0 & \ldots & \ldots & \ldots \\ a_1 & c_1 & b_1 & 0 & \ldots & \ldots \\ 0 & a_2 & c_2 & b_2 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & 0 & a_{I-1} & c_{I-1} & b_{I-1} \\ \ldots & \ldots & \ldots & 0 & a_I & c_I \end{pmatrix} \tag{3.22}$$

and

$$\boldsymbol{Q}_u^o = (0, \ldots, 0, (N-I)\gamma)^T, \tag{3.23}$$

where $a_i = i\alpha$, $b_i = (N-i)\gamma$, and

$$c_i = \begin{cases} -(a_i + b_i), & i = 0 \ldots I-1 \\ -a_i, & i = I \end{cases} \tag{3.24}$$

The initial probability vector defined by Eqs. (3.25) and (3.26) is approximated by the steady state probability of the number of active connections.

$$\omega_k = \begin{pmatrix} N \\ k \end{pmatrix} \left(\frac{\alpha^{-1}}{\alpha^{-1}+\gamma^{-1}}\right)^k \left(\frac{\gamma^{-1}}{\alpha^{-1}+\gamma^{-1}}\right)^{N-k} \tag{3.25}$$
$$k = 0, \ldots, I$$

$$\omega^o = \sum_{k=I+1}^{N} \begin{pmatrix} N \\ k \end{pmatrix} \left(\frac{\alpha^{-1}}{\alpha^{-1}+\gamma^{-1}}\right)^k \left(\frac{\gamma^{-1}}{\alpha^{-1}+\gamma^{-1}}\right)^{N-k}. \tag{3.26}$$

$\boldsymbol{Q}_u$ is irreducible if the system is stationary. The asymptotic behavior of this phase type distribution depends on the maximum eigenvalue of $\boldsymbol{Q}_u$ with the largest real part. That is, if $\boldsymbol{Q}_u$ is irreducible, this phase type distribution is asymptotically exponential [57]. The complementary probability distribution function is given by

$$F^C(x) = Ke^{-\eta x} + o(e^{-\eta x}), \qquad x \to \infty, \tag{3.27}$$

where $-\eta$ is the real eigenvalue of $\boldsymbol{Q}_u$, and $K = \boldsymbol{\omega}\boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is the positive right vector corresponding to $-\eta$. $\boldsymbol{\nu}$ is determined uniquely by the condition that $\boldsymbol{u}\boldsymbol{\nu} = \boldsymbol{u}\boldsymbol{e} = 1$, where $\boldsymbol{u}$ is the left eigenvector corresponding to $-\eta$. In general, solving a eigenvalue of an asymmetric matrix is difficult. Fortunately, $\boldsymbol{Q}_u$ is an essentially nonnegative matrix and therefore it is possible to solve the maximum real eigenvalue by using a power method or inverse iteration method [59, 1].

Stationary distribution of the number of connections

Without admission control, the stationary distribution of the number of connections is given by (see Fig. 3.14(a))

$$P_i = \prod_{k=1}^{i} \frac{\lambda}{k\mu} P_0 \tag{3.28}$$

$$P_0 = \left(\sum_{i=0}^{\infty} \prod_{k=1}^{i} \frac{\lambda}{k\mu} \right)^{-1}. \tag{3.29}$$

With admission control, the stationary distribution of the number of connections is given by (see 3.14(b))

$$P_i = \prod_{k=1}^{i} \frac{\lambda\gamma_{k-1}}{k\mu} P_0 \tag{3.30}$$

$$P_0 = \left(\sum_{i=0}^{\infty} \prod_{k=1}^{i} \frac{\lambda\gamma_{k-1}}{k\mu} \right)^{-1}. \tag{3.31}$$

The γ_{k-1} in Eqs. (3.30) and (3.31) is the multiplication factor for the upward transition of the birth-and-death process; it depends on the admission control algorithm. In the case of admission control with fixed threshold K,

$$\gamma_k = \begin{cases} 1 & \text{if } k < K \\ 0 & \text{otherwise.} \end{cases} \tag{3.32}$$

Threshold K is pre-computed by assuming the complete traffic characteristic in advance [60, 52, 61, 1]. The γ_k is given by Eq. (3.20) for our proposed method. Using Eq. (3.20), (3.30), and(3.31) gives us a distribution of the number of connections admitted with our proposed method.

Numerical example

The relationship between the average number of admitted connections and the mean connection-holding time for various monitoring periods is shown in Fig. 3.15, assuming that λ^{-1}=0.01 [sec], α^{-1}=0.1 [sec],β^{-1}=10 [kBytes], R=6.3[Mb/s].

As the mean connection-holding time becomes longer, the average number of admitted connections increases and becomes saturated. In other words, even if the mean connection-holding time were infinite, there is an

Table 3.2: Relationship between the target CLR and the admissible number of connections

CLR	number of connections
1.0e-7	67
1.0e-6	76
1.0e-5	86
1.0e-4	99
1.0e-3	117

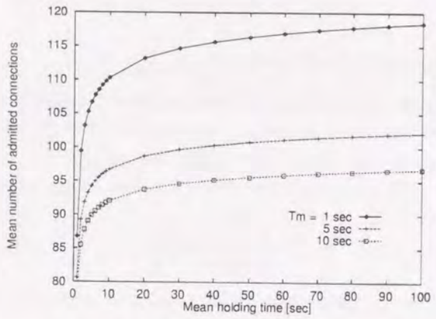


Figure 3.15: Relationship between average number of admitted connections and mean connection-holding time.

upper limit to the number of connections admitted by our proposed method. This confirms that our proposed method regulates the number of connections properly. The level at which the average number of connections becomes saturated is higher, the shorter the monitoring period. The shorter the monitoring period, the easier it is for a connection to be accepted.

Table 3.2 shows the relationship between the number of admitted connections and the target CLR. The CLR was calculated using the buffer-less fluid flow model [52]. From Table 3.2 and Fig. 3.15, $T_m = 10$ yields a CLR between 1.0e-4 and 1.0e-5.

3.5.2 Comparison with conventional CAC methods

By computer simulation, we compare performance of our proposed method and conventional CAC methods [11].

Comparison with conventional traffic descriptor based CAC methods

We evaluated the performance of our method through computer simulation. We modeled the cell, burst, and connection-level behavior. To investigate the robustness of our method, we assumed that the connection arrival rate is extremely high: the connection setup request arrival interval is set equal to one cell slot. We assumed that the connection holding time is infinite.

We assume that $C = 149.76$ Mbit/sec and that $R = 1.5$ or 10 Mbit/sec, $a = 20$, $B = 10$ Kbytes, $\alpha =$ 4.156e-3, $\lambda_{target} = 0.8$, and $T_m = 10$ sec. The number of admitted connections as a function of elapsed time is plotted in Fig. 3.16. Figure 3.16 also shows the number of connections admitted by the conventional traffic descriptor based CAC methods, i.e., the virtual cell loss probability [52] and the effective bandwidth [1].

The proposed method approaches to the level of the virtual cell loss probability method over time. The virtual cell loss probability takes into account the burst level behavior, and it calculates the knee point of the CLR-buffer curve assuming that excess burst arrivals are discarded (recall Fig. 3.1). The proposed method also takes into account the burst level behavior, and it regulates the admission of connections so that the cell rate of active connections does not exceed the link capacity for a sufficiently long period while maintaining the target CLR. The basic idea behind these two methods is similar, i.e., they suppress the occurrence of burst arrivals exceeding the VP capacity, allowing a statistical multiplexing gain between different connections. Note that the proposed method is measurement based while the virtual CLR method uses the user-declared connection's peak rate and average rate.

The proposed method efficiently admits more connections than the effective bandwidth method, as shown in Fig. 3.16. The reason is as follows. The proposed method anticipates the statistical multiplexing gain between different connections. Generally speaking, this approach achieves high efficiency when the connection's peak rate is small compared to the link capacity. In contrast, the effective bandwidth method does not exploit a statistical multiplexing gain between connections, so is too conservative except when the connection's peak rate is high and the buffer is sufficiently large (comparable to the connection's average burst size). Here we assume that the connection's peak rate and buffer size are small compared to the connection's burst size. Therefore, the proposed method yields a higher bandwidth efficiency than does the effective bandwidth method.

The proposed method admits around 80% (around 65%) of the number of connections admitted by the virtual cell loss probability method for connections with $R = 1.5$ Mbit/sec (10 Mbit/sec). Note that the number of connections admitted by the virtual cell loss probability method may become lower than that shown in Fig. 3.16 if the declared connection's average rate and average burst length contain the safety margin to offset their inaccurate estimations as mentioned in Section 3.1.

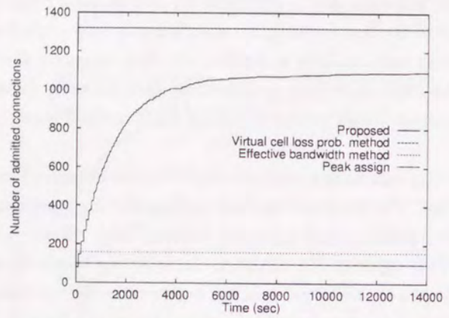
The efficiency at $R = 1.5$ and 10 Mbit/sec differ due to the statistical multiplexing effect. That is, an $R =$ 1.5 Mbit/sec is more readily acceptable for the same residual bandwidth than an $R = 10$ Mbit/sec.

Also as shown in Fig. 3.16, the proposed method more than triples the bandwidth obtained using the peak assignment method with an $R = 10$ Mbit/sec, while the increase is ten-fold for an $R = 1.5$ Mbit/sec. This advantage is due to use of direct traffic measurements. In the proposed method, the measurements are obtained using a simple LPF described in Section 3.4. If the average rate and average burst length cannot be accurately determined beforehand, the peak assignment method is the most suitable solution. Our proposed method is more effective for multimedia traffic with unknown traffic characteristics.

Comparison with the Dynamic CAC method

In order to compare performance of our proposed method and the Dynamic CAC [11], we conducted computer simulation. We used condition in Table 3.3. We varied arrival interval and holding time of connections.

Figure 3.17 shows that time series data of the number of connections admitted by our proposed method with various holding times of connections. We used $\lambda^{-1} = 0.01$ [sec], and R= 6.3 [Mb/s]. As the holding time increases, the average number of connections increases, approaching to asymptotic level. The asymptotic level



(a) Connection's peak rate = 1.5 Mbit/sec.

(b) Connection's peak rate = 10 Mbit/sec.

Figure 3.16: The number of admitted connections as a function of time. For the effective bandwidth method proposed by Elwalid and Mitra [1], the buffer size is set to 1000 cells.

Table 3.3: Simulation condition

Input traffic
$\lambda^{-1} = 0.001, 0.01, 0.05$ [sec]
$\mu^{-1} = 2, 8, 5, 30, 50$ [sec]
$\alpha^{-1} = 0.1$ [sec], $\beta^{-1} = 10$ [kBytes]
$R = 1.5, 6.3, 10, 15$ [Mb/s]
Our proposed method
$T_m = 1$ [sec]
Dynamic CAC [11]
$\alpha = 0.1$
window size=128 cells
renewal period = 128×1024 cells
CLR = 1.0e-6

is equal to an average of the number of connections obtained by the analytical model in Subsection 3.5.1.

In a region when the holding time is short, the average number of connections in Fig. 3.15 is larger than that in Fig. 3.17. This is because the analytical model does not take into account of effect of the conservative modification after a new connection is admitted (see Section 3.4.3) . The difference is, however, small when the holding time becomes longer because new arriving connections are likely to be accepted after the monitoring period.

Figure 3.17 shows that time series data of the number of connections admitted by our proposed method with various arrival interval of connections. We used $\mu^{-1} = 5$ [sec], and $R = 6.3$ [Mb/s]. There is little impact of the arrival interval on the average number of the connections. Because the conservative modification is performed in our proposed method, it avoids accepting too many connections even if the arrival intervals are very short. Thus, we claim that our proposed method is robust against bulk arrival of connections.

Next we compare the performance of our proposed method and that of the Dynamic CAC. We varied the connection's peak rate as in Table 3.3. Time series data of the number of connections are plotted in Fig. 3.19. We set $R = 6.3$ Mb/s, $\lambda^{-1} = 0.01$ [sec], and $\mu^{-1} = 5$ [sec]. Since traffic intensity of connection arrival amounts to 500 [Erl], the average number of connections would become 500 without admission controls. Our proposed method and the Dynamic CAC regulate the number of connections around 80, as shown in Fig. 3.19.

Time series data of the number of connections with various connection's peak rate R are plotted in Fig. 3.20. There is little relationship between the connection's peak rate R and the number of connections.

Figure 3.21 shows the relationship between the maximum number of connections and connection's peak rate. Our proposed method achieves the maximum number of connections comparable to that by the Dynamic CAC.

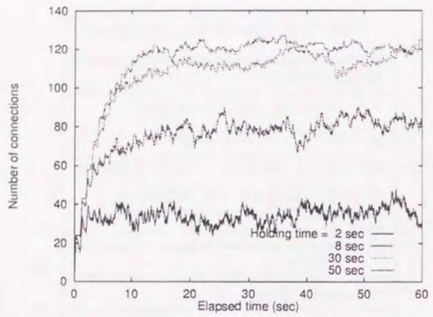


Figure 3.17: Time series data of the number of connections admitted by our proposed method with various holding times.

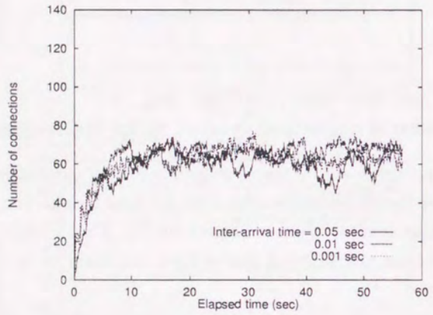


Figure 3.18: Time series data of the number of connections admitted by our proposed method with various arrival intervals.

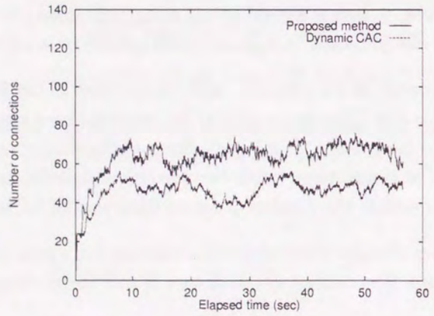
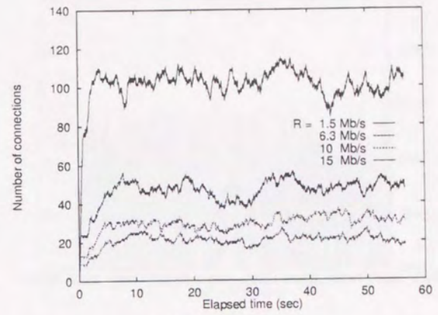
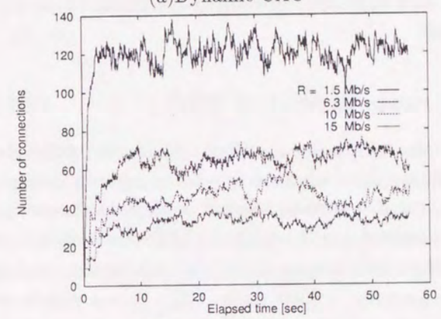


Figure 3.19: Time series data of the number of connections admitted by our proposed method and the Dynamic CAC.



(a)Dynamic CAC



(b)Proposed method

Figure 3.20: Time series data of the number of connections with various connection's peak rate by our proposed method and the Dynamic CAC.

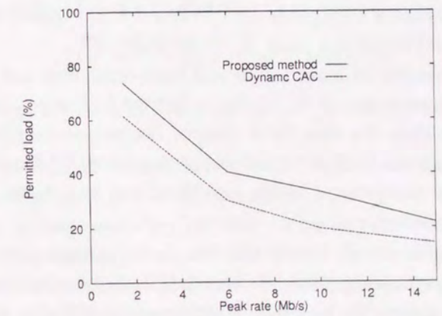


Figure 3.21: Maximum number of connections admitted by our proposed method and the Dynamic CAC as a function of connection's peak rate.

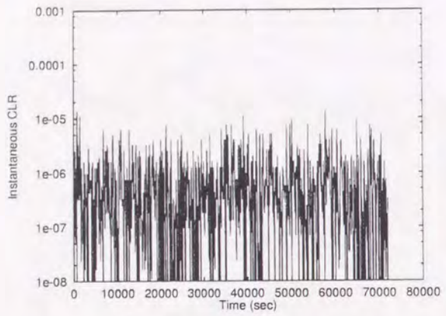


Figure 3.22: Instantaneous CLR with exponential burst length and inter-burst time. T_m = 120 [sec], T_{PCR} = 10 [sec], ξ = 600 [sec], η = 0.1 [sec]

3.5.3 Effect of the long range dependent traffic

Recently it has been reported that the LAN traffic and VBR video traffic exhibit long range dependence[38, 62]. Long range dependent (LRD) traffic has much more impact on network performance than the short range dependent (SRD) traffic such as Poisson process, Markov modulated Poisson process. It is worthwhile to examine the performance of our proposed method using the LRD traffic model.

It is known that a aggregate traffic from a large number of source traffic, of which burst length distribution exhibits long tail distribution, constitutes a LRD traffic [63]. Pareto distribution is known as a long tail distribution and fits well burst length of TCP traffic [39]. We assumed a burst length and inter-burst length distribution conform to Pareto distribution. Probability distribution function (PDF) of Pareto distribution is defined as

$$F(x) \equiv Prob[X \leq x] = 1 - (a/x)^{\beta}, \tag{3.33}$$

where $x \geq a$, $a, \beta \geq 0$ and β is called a scale parameter. When $\beta < 2$, variance is infinite. It is known that infinite variance of individual burst length is a cause of self-similarity [63].

We assumed that both distribution of burst length and inter-burst time are Pareto distribution with β = 1.2 (infinite variance). We assume that β = 1.2, T_m = 120 [sec], T_{PCR} = 10 [sec], ξ = 600 [sec], η = 0.1 [sec]. Figures 3.22 and 3.23 show the time series data of the instantaneous CLR with exponential and Pareto distribution. The instantaneous CLR is defined as the theoretical CLR associated with the number of connections. The associated CLR is calculated by the virtual cell loss probability [52] with the peak rate and burstiness factor used in the simulation.

From these figures, we observe that there is little difference in the network performance. It is reported that Pareto distribution results in hyperbolically decay of queue length distribution while exponential distribution in exponential decay. When we discuss the long time scale where the buffering effect matters, the difference might be significant. But in our proposed method we use the *instantaneous* rate as the admission criteria. The effect of long range dependent traffic is small in such a region. Thus we claim that LRD traffic does not degrade performance of our proposed method.

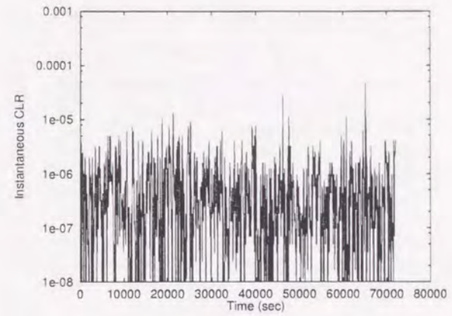


Figure 3.23: Instantaneous CLR with Pareto burst length and inter-burst time. β = 1.2, T_m = 120 [sec], T_{PCR} = 10 [sec], ξ = 600 [sec], η = 0.1 [sec]

3.6 Implementation

Because admission control is invoked when the connection is setup, the admission decision should be quick so as to minimize the connection setup delay. Namely, computational complexity in the admission decision procedure should be minimized. In addition, from the viewpoint of switching node cost, extra hardware and software control should be avoided. In this section we address the computational complexity in the admission decision and describe a simple hardware implementation of our proposed method.

3.6.1 Hardware implementation

As described in Section 3.4, our proposed bandwidth management method uses a LPF to obtain the instantaneous rate. The cells are counted and the instantaneous rate is calculated for every cell slot by using the LPF in Eq. (3.6). By expressing smoothing coefficient α as a power of two, i.e., 2^{-k}, we can avoid the floating point operations and thus implement the LPF as a combination of an adder, a subtracter, and two shifters:

$$\lambda(t) = 2^{-k} n(t) + (1 - 2^{-k})\lambda(t - \Delta). \tag{3.34}$$

As shown in Eq. (3.34) and Fig. 3.24, the LPF is composed of one adder, one subtracter, and two k-bit-right shifters.

In our proposed method, the residual bandwidth is derived from the maximum of the observed instantaneous rate. We therefore have to keep track of the instantaneous rate over the monitoring period. Because the maximum instantaneous rate is calculated every cell slot, we would need a tremendous amount of memory if we stored *all* the instantaneous rates during the monitoring period. Fortunately, only a few of them are required to calculate the *maximum* instantaneous rate. We reduce the storage capacity by dividing the monitoring period into n bins, as shown in Fig. 3.25. Cyclic queue $\Lambda[i]$ $(i = 0 \ldots n-1)$ contains maximum instantaneous rate associated with each bin. Two registers, λ and Λ , contain the current instantaneous rate and the maximum instantaneous rate among all bins, i.e., $\Lambda = \max_i \Lambda[i]$. Now suppose that the current cell slot is the first cell slot of the $(n-1)$-th bin. The instantaneous rate is calculated for the current cell slot and loaded into $\Lambda[n-1]$.

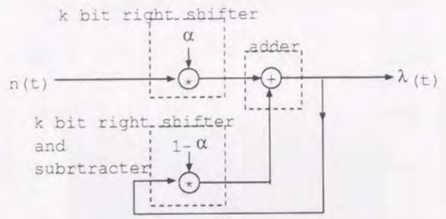


Figure 3.24: Block diagram of the low-pass filter.

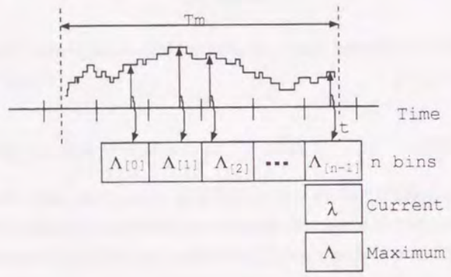


Figure 3.25: Division of monitoring period.

If it is larger than the value Λ, it is also loaded into Λ. In the next cell slot, i.e., the second cell slot of the $(n-1)$-th bin, the instantaneous rate is calculated again and checked to see if it is larger than $\Lambda[n-1]$ or not. If it is, it is loaded into $\Lambda[n-1]$. The current instantaneous rate is also checked to see if it is larger than Λ or not. If it is, Λ is replaced by the current instantaneous rate as well. The same procedure is repeated until the end of the bin is reached. At the end of the $(n-1)$-th bin, the maximum instantaneous rate is selected from among all $\Lambda[i]$, excluding $\Lambda[0]$, which is used to store the maximum instantaneous rate observed in the next bin. The implementation complexity depends on the number of bins.

The relationship between the average number of connections and the number of bins is shown in Fig. 3.26. The impact of the number of bins is quite small. A two-bin configuration achieves almost the same average number of connections as a ten-bins one. The complexity of the measurement process is thus considerably reduced while the average number of connections is maintained.

3.6.2 Computational complexity of the admission decision

The admission decision should also be quick to avoid long connection setup latency. We compare the computational complexity of our proposed method to that of Dynamic CAC.

In Dynamic CAC, the decision is made by comparing the target CLR to the estimated CLR by using the

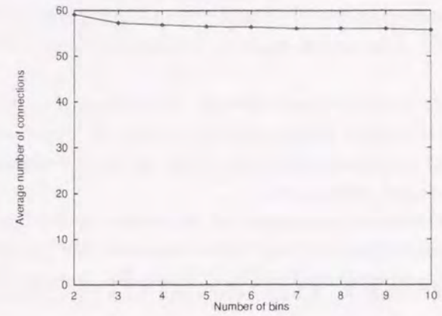


Figure 3.26: Relationship between number of admitted connections and number of bins, for an exponentially-distributed connection arrival interval (mean of 0.1 sec) and a holding time (mean of 600 sec), $T_m = 10$ sec, $\lambda_{target} = 0.8$, $R = 10$ Mbit/sec, $a = 20$, $B = 10$ Kbytes.

upper bound formula for the cell loss ratio:

$$CLR(t) = \frac{\sum_{k=\max(K,R)}^{M+R}(k-K)\,\hat{a}(k-R;t)}{\sum_{k=R}^{M+R} k\,\hat{a}(k-R;t)}, \tag{3.35}$$

where K denotes the buffer size, and M denotes the maximum number of cells arriving within a window whose size is K cell transmission times on a link. If the ATM switching fabric has N input ports, M=NK. Even though the denominator in Eq. (3.35) is equal to the average number of arriving cells within the window, the numerator requires approximately $3K(N$-$1)$ floating point operations. Our proposed method admission decision procedure, described in Section 3.4.2 is simpler. Admitting a new connection is simply judged by comparing $\lambda_{max}(t)$+R to λ_{target}, which requires only two operations.

The computational complexities of our proposed CAC and Dynamic CAC are plotted in Fig. 3.27. We assume that the buffer size $K = 128$ cells. Dynamic CAC requires hundreds of operations to calculate the cell arrival distribution and a DSP to calculate the CLR. In contrast, our proposed CAC, which incurs much less computational complexity, can be implemented with only a couple of registers. Compared to the conventional admission control method, our proposed method is suited for real-time processing, a necessity in practical ATM switching systems. Because our proposed method can be implemented with reasonable hardware complexity and the admission decision is fast, it is better suited for practical ATM switching systems.

3.7 Concluding remarks

In this chapter we presented the design for a low-pass filter that calculates the instantaneous rate and a framework for bandwidth management based on the low-pass filtered instantaneous rate. The recursive type LPF was shown to perform stably regardless of the connection's peak rate. The bandwidth management framework is based on the maximum instantaneous rate observed during the monitoring period. Analysis of the distribution of the underload periods, in which the instantaneous rate does not exceed the link capacity, showed that the monitoring period is affected significantly by the mean burst length. Analysis of a distribution

of the number of admitted connections showed that as the mean holding time increases, the average number of admitted connections increases and becomes saturated. Our proposed method thus regulates the number of connections properly.

We compared the performance of our proposed method and to that of conventional CAC methods and found that our proposed method achieves a higher maximum number of connections.

We investigated the impact of long range dependent traffic on the performance of our proposed method and found that it does not degrade the performance.

We demonstrated a simple hardware implementation of our method by dividing the monitoring period into several bins and expressing the smoothing coefficient of the recursive LPF by a power of two. We showed that the computational complexity of our proposed method is low and constant irrespective of the size of the switching fabric.

Because our bandwidth management method does not assume any mathematical model and can be implemented in simple hardware, it is well suited for practical ATM switching systems. This bandwidth management method should be extended to a network-wide bandwidth management framework. The design of such a framework is our future work.

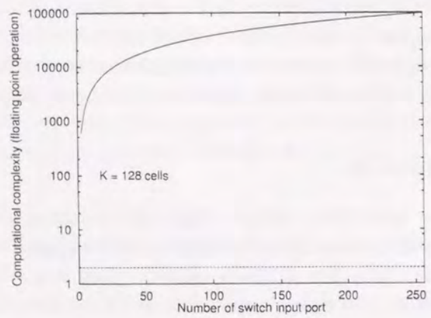


Figure 3.27: Comparison of the required computational complexity.

Chapter 4

An adaptive admission control for multiple QoS targets

We present simple and efficient connection admission control for multiple Quality of Service (QoS) requirements with respect to cell loss ratio (CLR). It alternates separate control, in which multiple queues are used to separate the bandwidth between the QoS classes in a deterministic fashion, and aggregate control, in which individual CLRs are used. The decision as to which control should be used is based on the number of connections, specifically, if weak-burst connections are requesting a higher QoS (a lower CLR) than that of the other connections, and if they are a small fraction of the total traffic volume, separate control is used; otherwise, aggregate control is used. This adaptive admission decision strategy maximizes the number of admitted connections. For separate control, a binary-search algorithm and a history of the number of admitted connections are used to allocate the bandwidth to each queue. This reduces the time it takes to re-allocate the bandwidth when a new connection is admitted or an existing one departs. For aggregate control, two fast algorithms are used to calculate the individual CLRs. Because the proposed method is simple and fast and achieves high bandwidth efficiency for a multiple QoS environment, it is suitable for future multimedia ATM networks.

4.1 Introduction

In ATM networks, connection admission control (CAC) determines whether a new connection setup request should be accepted or not by taking into account the current status of the network resource, the traffic descriptor and target Quality of Service (QoS) for the connection [8, 9]. Most of CAC methods that have been proposed, for example [60, 52, 64, 1, 65], are designed to user single QoS criteria. Few methods have been put forth for multiple QoS targets. Because ATM networks are designed to handle various kinds of media, we need to design CAC methods that can handle various combinations of QoS targets and traffic types [17, 66]. The first objective is to maximize the number of connections while satisfying the target QoS for each traffic type. Furthermore because CAC is performed in response to a connection setup request, the delay time due to CAC should be minimized so as to speed up the setup process. And finally, CAC should be simple enough to avoid expensive hardware and software.

We have developed a simple and efficient CAC method that can handle multiple QoS targets. Since it uses

one of a *separate* and an *aggregate* control adaptively, it achieve high bandwidth efficiency. It can be performed within a short period and does not delay the connection setup. In addition, since it takes less computational complexity than conventional methods, it does not require expensive hardware and software.

The rest of this chapter is organized as follow. In Sect. 4.2 we present the separate control. We first describe separate control, them aggregate control. In Sect. 4.3, we discuss the performance evaluation of our proposed method. In Sect. 4.4, we summarize this chapter and address our future work.

4.2 Adaptive admission control for multiple QoS targets

The proposed method uses one of two controls adaptively: a *separate* control, in which separate queues are used to separate the bandwidth between QoS classes in a deterministic fashion, and an *aggregate* control, in which an individual cell loss ratio (CLR) is used for each QoS class. The decision as to which control should be used is based on the number of connections. That is, if weak-burst type connections are requesting a higher QoS (a lower CLR) than that of the other connections, and if they are a small fraction of the total traffic volume, separate control is used; otherwise, aggregate control is used. Due to this adaptive admission decision strategy, we can maximize the number of admitted connections. For separate control we use a binary-search algorithm and a history of the number of admitted connections to determine the bandwidth allocation for each queue so as to reduce the time needed for bandwidth re-allocation when a new connection is admitted. For the aggregate control, we use two fast algorithms for calculating the individual CLRs so as to to avoid a huge amount of computational complexity.

4.2.1 Separate control

As for the separate control, cell-level performance is investigated in the literatures [67, 68, 69, 70]. The required bandwidth to maintain the target CLR is determined by the peak and average rates and the number of connections. We need to calculate the required bandwidth when the number of connections changes. Under this situation, bandwidth is over-allocated to the queue to avoid performance degradation until the accurate bandwidth allocation is determined and so bandwidth efficiency may degrade during this period. We develop a binary-search algorithm for calculating the required bandwidth to reduce the bandwidth calculation time. In addition, we present a idea to avoid the bandwidth calculation by using a history of the number of admitted connections.

Mechanism for guaranteeing bandwidth

Figure 4.1 shows a mechanism for guaranteeing bandwidth for separate control. Each queue is implemented in shared buffer memory [71], and bandwidth is allocated by scheduling the sequence of queues. We assume here that the cyclic scheduling pattern is pre-configured and stored in the scheduling table [67, 72]. The queue is cyclically selected based on the sequence pattern stored in the scheduling table. Each entry in the scheduling table holds the queue identifier to be selected at the time slot. When there are no cells in the queue to be served, there are two choices: either simply skip that time slot or select another queue. Selecting another queue improves cell-level performance, but we cannot rely on this for increasing the maximum number of admitted

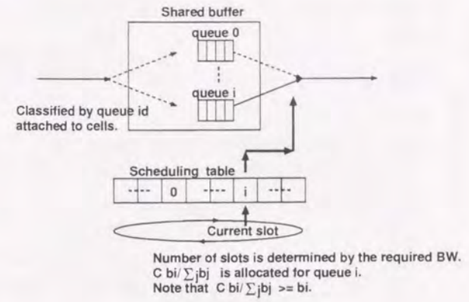


Figure 4.1: Mechanism for guaranteeing bandwidth

connections because there is not always an empty queue at each time slot. Therefore, there are no distinction between these two choices.

Bandwidth is allocated to each queue in such a a way the minimum bandwidth needed to maintain the target CLR is allocated and no bandwidth is left over. Namely, $Cb_i/\sum_j b_j$ is allocated to a queue of class i, where C and b_i denote the link capacity and required bandwidth for class i. If we allocate b_i to class i, C-$\sum_j b_j$ is left over. Introducing a mechanism that would enable leftover bandwidth to be shared by all queue would improve the cell-level performance. Unfortunately, such a mechanism would be complicated because it need a sophisticated algorithm to resolve contention for the shared resource. In contrast, our bandwidth allocation mechanism does not leave any bandwidth left over, simplifying the multiple queue configuration. We should note that C-$\sum_j b_j$ is considered to be un-allocated with respect to the CAC.

Required bandwidth

As mentioned, the bandwidth required to maintain the target CLR is determined by the peak and average rates and the number of connections. That is, given peak rate R_i, average rate a_i, target CLR CLR_i, and number of connections N_i, the required bandwidth is

$$b_i(N_i) = \min_c\{c : g(R_i, a_i, N_i, c) \leq CLR_i\}, \tag{4.1}$$

where $g(R_i, a_i, N_i, c)$ denotes the cell loss ratio function. Letting $f(x)$ and $F^c(x)$ denote the probability density function and the complementary function for the cell rate over all the aggregate connections, we have the CLR for link capacity C,

$$CLR = \frac{1}{A}\int_0^\infty (x-C)^+ f(x)dx \tag{4.2}$$

$$= \frac{1}{A}\int_C^\infty F^C(x)dx, \tag{4.3}$$

where$(x)^+ \equiv \max(x, 0)$ and A denotes the sum of average rates of all the multiplexed connections. The relationship between the number of connections N_i and the required bandwidth $BWi(N_i)$, is plotted in Fig. 4.2 for three types of traffic. In figure, Type 1, 2, and 3 indicate the traffic types whose peak rate R and burstiness

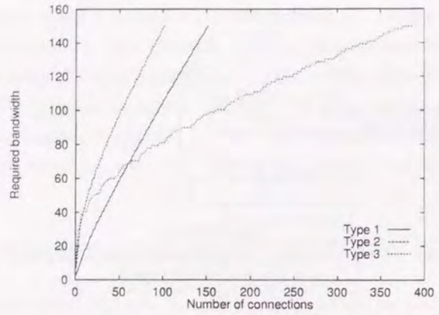


Figure 4.2: Relationship between number of connections N_i, and required bandwidth $BWi(N_i)$. Peak rate of Type 1, 2, and 3 connectoins are 1.5, 6, and 10 Mb/s.

factor a are 1.5 Mb/s and 2, 6 Mb/s and 10, 10 Mb/s, 100, where we define the burstiness factor is defined as the ratio of the peak to the average rates. The granularity of the bandwidth is 1.4976 Mb/s.

From Fig. 4.2, we observe that

- Traffic types 1 and 2 have knee points in the region where the number of connections is from 10 to 20. Type 3 has a knee point at around 150 connections.
- The required bandwidth is linearly related to the number of connections when the number of connections is larger than that at the the knee point.

Binary search algorithm for calculating required bandwidth

Because there is no explicit formula for calculating the required bandwidth, we estimate the required bandwidth by using the CLR calculation formula in a trial-and-error fashion. In general, calculating the CLR is a time-consuming operation. We should thus reduce the number of calculations needed as much as possible.

We developed a binary-search algorithm for this purpose. The required bandwidth is a monotonic decreasing function of the CLR. In our binary-search, the algorithm starts by selecting two points between which the target value is located. The following steps are performed recursive until the halt condition is satisfied. The halt condition is when the difference between the two points is less than the granularity of the bandwidth. If the function value corresponding to the center position between two points is larger than the target value, the center point and a point whose function value is smaller are selected as the next two points. Otherwise, the center point and a point whose function value is larger are selected.

The C++ code for the binary-search algorithm is shown in Fig. 4.3, which shows the interface and implementation for class `vbr`. Class `vbr` has peak rate, average rate, and the target CLR as private data `PCR`, `SCR`, and `QOS_CLR` (lines 4–6). The `req_bw` (line 9) is a class method for calculating the bandwidth required to maintain the target CLR when `nvc` connections are multiplexed; it calls recursive binary-search method `bw_bisec` (lines 13–15). Binary-search method `bw_bisec` calls itself recursively according to whether the center point is higher than the target or not. Because the required bandwidth should be smaller than the total link capacity,


```
1 //      vbr.H header file
2 class vbr {
3 private:
4   double PCR;          // peak bit rate [bits/sec]
5   double SCR;          // average bit rate [bits/sec]
6   double QoS_CLR;          // QoS objective regarding CLR
7 public:
8   double clr(double VPcap, int nvc);              // calculate CLR
9   double req_bw(double grn, int nvc);             // calculate required BW
10  double bw_bisec(double C0, double C1, double grn, int nvc); // binary search
11 };
12
13 //      vbr.C implementation file
14 double vbr::req_bw(double grn, int nvc)
15 { return(this.bw_bisec(0.0,PCR*nvc,grn,nvc));}
16
17 double vbr::bw_bisec(double C0, double C1, double grn, int nvc)
18 {
19   double ceil(double C, double grn);
20   double floor(double C, double grn);
21   C1 = ceil(C1,grn);     // C1 must support QoS_CLR
22   C0 = floor(C0,grn);    // C0 must not support QoS_CLR
23   if (C0 == C1-grn) return(C1);  // C1 is the lowest BW supporting QoS_CLR
24   else {                                                  // bi-section search
25     double C2 = (C0+C1)/2.0;
26     C2 = ceil(C2,grn);
27     if (this.clr(C2,nvc) < QoS_CLR) C1 = C2;          // select lower half
28     else C0 = C2;                                     // select upper half
29     return(bw_bisec(C0,C1,grn,nvc));
30   }
31 }
32
33 double ceil(double value, double grn)
34 { return(ceil(value/grn)*grn);}
35
36 double floor(double value, double grn)
37 { return(floor(value/grn)*grn);}
```

Figure 4.3: C++ code for binary-search algorithm for calculating the required bandwidth.

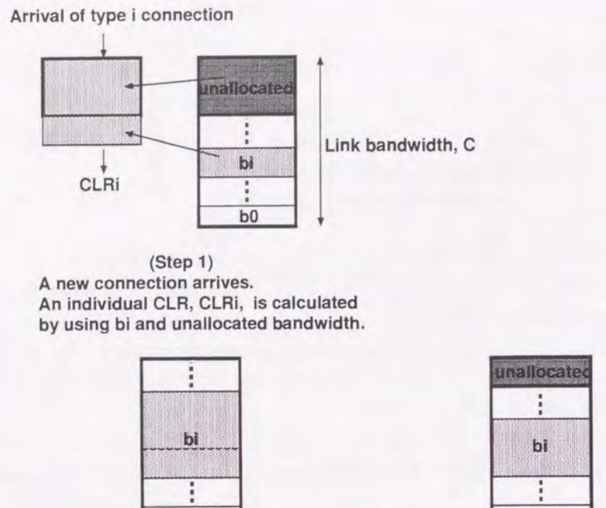


Figure 4.4: Bandwidth re-allocation method for separate control.

the link capacity is selected as the point that gives a CLR lower than the target CLR. Zero bandwidth is selected as the point that gives a CLR higher than the target. Method `req_bw` calls method `bw_bisec` after setting these two points as the initial two points. Given the granularity of bandwidth `grn`, the two points are rounded so that there is a target point between them (lines 21–22).

The computational required by this algorithm is determined by the ratio of the bandwidth granularity Δ_b to the total link bandwidth. The CLR may be calculated upto $\lceil \log_2 C/\Delta_b \rceil$ times until the required bandwidth is determined. The bandwidth granularity is determined by hardware implementation of multiple queue and schemduling mechanism. For the implimentation example Fig. 4.1, the width of the scheduling table determines the bandwidth granularity. If the width is 100 cell slots, the bandwidht granurarilty Δ_b is $C/100$ and the number of CLR calculations is 7 ($=\lceil \log_2 100 \rceil$). We can reduce this number by giving the current bandwidth as one of two points when method `req_bw` calls binary-search method `bw_bisec`.

Bandwidth re-allocation method

Figure 4.4 shows bandwidth re-allocation method. When a new connection arrives, the bandwidth already allocated to the class of that connection plus the unallocated bandwidth is used to calculate the CLR. For example, if a class i connection arrives, we assume that $C - \sum_{j \neq i}^{j} b_j$ is re-allocated to class i, and calculate

the CLR. If the CLR is lower than the target value, the connection is accepted: otherwise it is rejected. The unallocated bandwidth is temporarily used for the admission decision for a newly arriving connection.

If the newly arriving connection is accepted, $C - \sum_{j \neq i}^{j} b_j$ is re-allocated to class i. The sequence of queues is re-scheduled accordingly. In this case, bandwidth is over-allocated to class i. Then, the minimum required bandwidth is calculated. The sequence of queues are then re-scheduled. When a existing connection departs, the minimum required bandwidth is calculated. Again the sequence of queues is then re-scheduled.

Consequently, when the number of connections changes, the minimum required bandwidth needs to be calculated. Until then, the bandwidth is over-allocated to the queue. Because bandwidth is not allocated to other queues during this period, bandwidth fficiency may be lowered. To avoid this problem, we pre-compute the required bandwidth for the current number of connections plus and/or minus one connection when the CAC processor is idle. The situation where the number of connections is larger or smaller than the current one by one must have been previsouly calculated. This can be done effectively, by storing the history of the previous states, i.e., the required bandwidths. Especially when all the leftover bandwidth is allocated to the class of a newly arriving connection, it is effective to store the required bandwidth corresponding to the current number of connections plus one.

4.2.2 Aggregate control

Individual CLRs

In aggregate control, traffic types are classified based on their peak and average rates. When connections of different types are multiplexed into a single queue, the aggregate CLR is different from the individual CLRs observed by the individual connections [52]. This implies that there is a possibility that multiple QoS classes can be established within a single queue. Assuming that there is no priority control mechanism in the single queue, cell loss rate of a certain connection is proportional to the ratio of the cell rate for that connection to the total cell rate for all the connections. Thus the individual CLR for type i is given by

$$CLR_i = \frac{1}{N_i a_i} \int_0^\infty (z - C)^+ \int_0^\infty \frac{x}{z} f_i(x) f_{\bar{i}}(z - x) dx dz, \tag{4.4}$$

where $f_i(x)$ denotes the probability density function (pdf) for the cell rate for all connections of type i, $f_{\bar{i}}(x)$ denotes the pdf for the cell rate for all connections except connections of type i, and N_i and a_i denote the number of connections and the average rate of type i connection.

Figure 4.5 shows how the individual and aggregate CLRs change with change in the mix of traffic types. Type 1 traffic is strong-burst traffic ($R_1 = 10$ Mb/s, and $a_1 = 0.01$), while type 2 traffic is weak burst-traffic ($R_2 = 1.5$ Mb/s, $a_2 = 0.5$). We refer to traffic with a high burstiness factor as a *strong*-burst type and to traffic with a low burstiness factor a *weak*-burst type. The horizontal axis indicates the number of type 1 connections, and the vertical axis indicates the aggregate CLR and the individual CLRs for both traffic types. The number of type 2 connections is determined such that both individual CLRs do not exceed the threshold value, 1.0e-6. In Fig. 4.5, CLR, CLR1, and CLR2 denote the aggregate CLR, the individual CLR for type 1, and the one for the type 2. The relationship between the number of type 1 and type2 connections is shown in Fig. 4.7.

From Fig. 4.5, we observe that:

(1) CLR 1 is always larger than CLR 2.

(2) When the number of type 1 connections increases from 0 to 1, the CLR 2 drops sharply from 1.0e-6 to 1.0e-8.

(3) As the number of type 1 connections increases, CLR 2 increases. It increases especially sharply when the number of type 1 connections is smaller than 50.

(4) The aggregate CLR is close to the CLR 2 when the number of type 1 connections is small, while it is close to the CLR 1 when the number of type 1 connections is large.

(5) When the number of type 1 connections is 382, the number of type 2 connections is 0, so CLR 2 is 0.

The reason for observation (1) is that type 1 connections are more likely to encounter congestion than type 2 connections. That is, a bursty connection contributes to the congestion when it sends cell, as shown in Fig. 4.6. Observation (2) implies that the number of type 2 connection is limited when there are also type 1 connections. The reason for observation (3) is that when the number of type 1 connections is small, type 1 connections are more likely to encounter congestion than type 2 ones. As the number of type 1 connections increases, the type 2 connections becomes more likely to encounter congestion, so the difference between the two individual CLRs becomes smaller.

From Fig. 4.7, we observe that the followings:

(1) The number of type 2 connections drops more sharply with an increase in the number of type 1 connections when the number of type 1 connections is small.

(2) The relationship is approximately linear when the number of type 1 connections is between 150 and 330.

The reason for observation (1) is that when the number of type 1 connections is small, the CLR 2 increases sharply. The reason for observation (2) is that the region in which the number of connections is between 150 and 330 corresponds to the region after the knee point for the required bandwidth curve (see Fig. 4.2), which is where the statistical multiplexing effect is fully utilized.

Figure 4.8 shows the relationship between the number of type 1 connections and the bandwidth efficiency, where the bandwidth efficiency is defined as the ratio of the sum of the average rates to the total link capacity. Bandwidth efficiency degrades as the number of type 1 connections increases because type 1 connections are more bursty than type 2 ones.

Admission decision

Our algorithm using the individual CLRs is illustrated in Fig. 4.9. We have to evaluate the individual CLRs for all the types, in order to determine whether a connection of a certain type will cause interference between connections of different traffic types. Only if the individual CLRs for all the types are satisfied, is the connection accepted.

Approximations for individual CLRs

When the number of traffic types is K and the number of type i connections is N_i, the computational complexity for calculating the aggregate CLR is [73]

$$O((K + \sum_{i=1}^{K} N_i) \prod_{i=1}^{K} N_i). \tag{4.5}$$

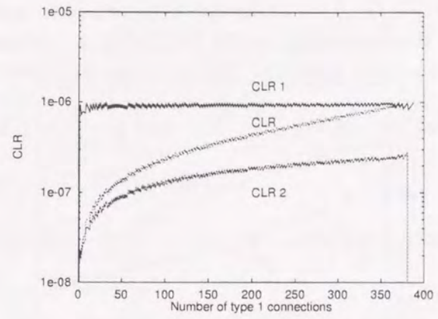


Figure 4.5: Relationship between the number of type 1 connections and the aggregate CLR and the individual CLRs.

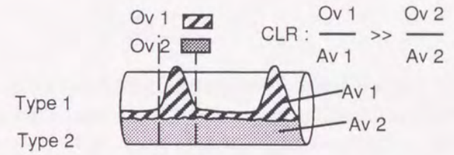


Figure 4.6: Bursty traffic is likely to see congestion.

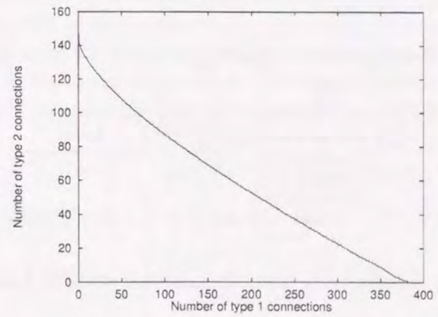


Figure 4.7: Relationship between the number of type 1 connections and the number of type 2 connections.

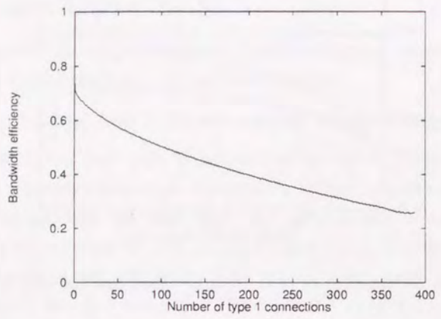


Figure 4.8: Relationship between the number of type 1 connections and the bandwidth efficiency.

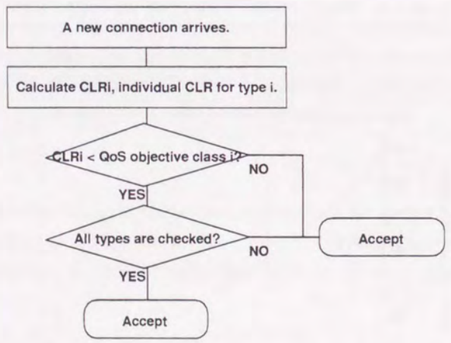


Figure 4.9: Algorithm using the individual CLR within a single queue.

When a new connection setup request arrives, we need to calculate the individual CLRs for all the types, which leads to a huge amount of computation if we use Eq. (4.4).

To reduce the computational complexity, we propose two approximations in what follows. The idea behind the first approximation is to multiply the aggregate CLR by a safety margin. Let us recall Eq. (4.4):

$$CLR_i = \frac{1}{N_i a_i} \int_0^\infty (z-C)^+ \int_0^\infty \frac{x}{z} f_i(x) f_i(z-x) dx dz.$$

When $z - C \geq 0$, it holds that $1/z \leq 1/C$. We thus have

$$CLR_i \leq \frac{1}{N_i a_i} \frac{1}{C} \int_0^\infty (z-C)^+ \int_0^\infty x f_i(x) f_i(z-x) dx dz.$$

Because $x \leq N_i R_i$, we have

$$\begin{aligned} CLR_i &\leq \frac{R_i}{a_i} \frac{1}{C} \int_0^\infty (z-C)^+ \int_0^\infty f_i(x) f_i(z-x) dx dz \\ &= \frac{R_i}{a_i} \frac{1}{C} \int_0^\infty (z-C)^+ f(z) dz \\ &= \frac{R_i}{a_i} \frac{1}{C} \int_0^\infty F^C(z) dz \\ &= \frac{R_i}{a_i} \frac{A}{C} CLR. \end{aligned} \tag{4.6}$$

This equation shows that the individual CLR is upper-limited by multiplying the aggregate CLR by the safety margin for the corresponding traffic type. That is, we can approximate the individual CLR by calculating the aggregate CLR and multiply it by $\frac{R_i}{a_i} \frac{A}{C}$. Note that the aggregate CLR is calculated with a computational complexity of $O(1)$ by using Chernoff upper-bound [73]. The computational complexity can thus be reduced dramatically by using Eq. (4.6). In this chapter, we use the Chernoff bound to calculate the aggregate CLR. We refer the method based on Eq. (4.6) as the Simplest method.

The second approximation is obtained by observing the individual CLR from another viewpoint. Suppose that connection i belongs to a certain type. We assume that there is no difference between CLRs of the same type, so we need consider the CLR of only connection i. Only when connection i sends cells, does cell loss occurs in connection i. Consequently, the individual CLR for connection i is given by Eq. (4.7). Let $f_{\tilde{i}}$ denote the pdf for the cell rate over all the connections, except connection i. Suppose that connection i sends cells. Only when the cell rate over all the connections except connection i exceeds $C - R_i$, does cell loss occur. We thus have

$$CLR_i = \frac{1}{R_i} \int_0^\infty (x + R_i - C)^+ \frac{R_i}{x + R_i} f_{\tilde{i}}(x) dx. \tag{4.7}$$

Because $R_i/(x + R_i) \leq R_i/C$ when $x + R_i - C \geq 0$, we have

$$\begin{aligned} CLR_i &\leq \frac{1}{R_i} \frac{R_i}{C} \int_0^\infty (x + R_i - C)^+ f_{\tilde{i}}(x) dx \\ &= \frac{1}{C} \int_0^\infty (x + R_i - C)^+ f_{\tilde{i}}(x) dx \\ &= \frac{1}{C} \int_{C_{\tilde{i}}}^\infty F_{\tilde{i}}^C(x) dx, \end{aligned} \tag{4.8}$$

where C_i denotes the total link capacity minus the peak rate of connection i.

Although we need to calculate the complementary distribution functions K times in Eq. (4.8), they can be calculated with a computational complexity of $O(1)$ [73]. We refer the method based on Eq. (4.8) as the Chernoff method.

Accuracy of approximations

We investigated the accuracy of those two approximations by using the same number of type 1 and 2 connections as in Fig. 4.5. Figures. 4.10, 4.11, and 4.12 show the approximated aggregate CLRs, CLRs for type 1 connections, and CLRs for type 2 connections. The Chernoff approximations were obtained using Eq. (4.8) and the simplest approximations were obtained using Eq. (4.6).

For Fig. 4.10, the Simplest method (Eq. (4.6)) is the same as the Chernoff method (Eq. (4.8)). The approximated aggregate CLR is higher than the exact CLR by about an order of magnitude, due to the Chernoff bound.

From Fig. 4.11 we observe that

(1) CLR 1 obtained using the Chernoff method (Eq. (4.8)) is higher than the exact CLR.

(2) The difference between CLR 1 obtained using the Simplest method (Eq. (4.6)) and that using the Chernoff method (Eq. (4.8)) is small when the number of type 1 connections is small. It increases with the number of type 1 connections.

(3) CLR 1 obtained using the Simplest method (Eq. (4.6)) is higher than that using the Chernoff method (Eq. (4.8)) by more than an order of magnitude when the number of type 1 connection is large.

(4) When the number of type 1 connections is larger than 381, CLR 1 using both methods are the same.

The reason for observation (1) is the safety margin of the Chernoff upper bound. The reason for observation (2) is as follows. When the number of type 1 connections is small, the aggregate CLR is smaller than CLR 1. It follows that the safety margin multiplier in Eq. (4.6) ($\frac{R_i}{a_i}\frac{A}{C}$) and the safety margin due to the Chernoff upper-bound are canceled. As the number of type 1 connections increases, the aggregate CLR approaches the CLR 1 and the cancellation effect decreases. The reason for observation (3) is that the safety margin multiplier is the order of magnitude of one. The reason for observation (4) is that the number of type 2 connections is zero.

From Fig. 4.12 we notice that CLR 2 obtained using the Simplest method (Eq. (4.6)) is a little higher than that obtained by the Chernoff method (Eq. (4.8)). This is because Eq. (4.6) yields a smaller safety margin multiplier. Note that $\frac{R_i}{a_i}$=100 for type 1 connections but only $\frac{R_i}{a_i}$=10 for type 2 ones.

The relationships between the maximum number of type 2 connections admitted using the exact method (Eq. (4.4)), the Simplest method (Eq. (4.6)), and the Chernoff method (Eq. (4.8)) and the number of type 1 connections are shown in Fig. 4.13. The constraint is that the individual CLRs for both types must be less than 1.0e-6. CLR 1 is always larger than CLR 2 except when the number of type 1 connections is zero.

From Fig. 4.13, we observe that

(1) When the number of type 1 connections increases from 0 to 1, the number of type 2 connections admitted using the Chernoff method (Eq. (4.8)) is smaller than that using the exact calculation. Also the number

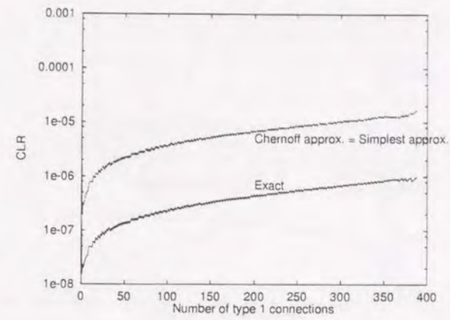


Figure 4.10: Approximated and exact aggregate CLRs.

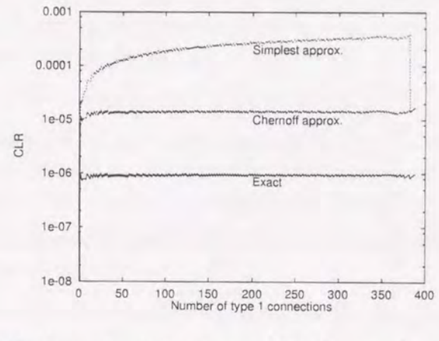


Figure 4.11: Approximated and exact CLRs for type 1 connections.

of type 2 connections admitted using the Simplest method (Eq. (4.6)) is smaller than that using the Chernoff method (Eq. (4.8)).

(2) When the number of type 1 connections is small, the difference between the number of type 2 connections admitted using the Simplest method (Eq. (4.6)) and that using the Chernoff method (Eq. (4.8)) is small.

The reason for observation (2) is as follows. The number of type 2 connections is limited by the type 1 connections. When the number of type 1 connections is small, the difference between CLR 1 using the Simplest method (Eq. (4.6)) and that using the Chernoff method (Eq. (4.8)) is small for those connections as shown in Fig. 4.11.

4.3 Performance evaluation

We evaluated the number of connections admitted by separate control and by aggregate control. The traffic types and QoS classes we used are listed in Tables 4.1 and 4.2.

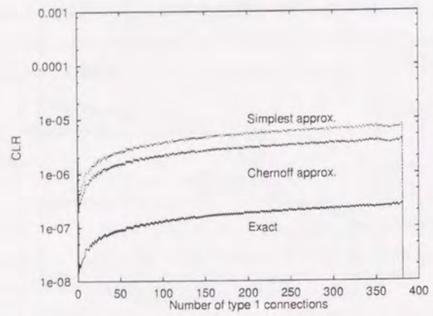


Figure 4.12: Approximated and exact CLRs for type 2 connections.

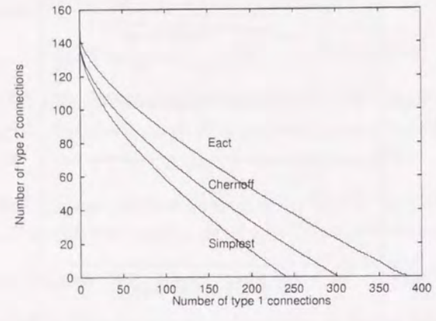


Figure 4.13: Number of type 2 connections admitted by exact calculation and by approximations.

Table 4.1: Traffic types

Type	Peak rate, R	Burstiness factor, a	Applications
(1) Data	10 Mb/s	100	Database lookup
(2) Voice	1.5 kb/s	2	Hi-fi telephony
(3) Video	6 Mb/s	10	TV conferencing

Table 4.2: QoS classes

Class	QoS
High CLR	1.0e-5
Low CLR	1.0e-7

Table 4.3: Combinations of type and QoS class for two-types case.

Type 1 \ Type 2	High CLR	Low CLR
High CLR	match 1	match 2
Low CLR	match 3	match 4

4.3.1 Two-type case

First we considered the case in which two traffic types, type 1 (data) and type 2 (voice), coexist, evaluating the number of connections admitted for the combinations of type and QoS class shown in Table 4.3.

The number of type 2 connections admitted by separate and aggregate controls as a function of the number of type 1 connections for match 1, 2, 3, and 4 are shown in Figs. 4.14, 4.17, 4.20, and 4.21.

In Fig. 4.14, both a strong-burst type 1 connection and a weak-burst type 2 connection are requesting the same low QoS level. From this figure we observe that

(1) Aggregate control outperforms separate control for all the region.

(2) With aggregate control, when the number of type 1 connections increases from 0 to 1, the number of type 2 connections drops sharply from 152 to 141. In contrast, when the number of type 2 connections increases from 0 to 1, the number of type 1 connections drops slightly from 370 to 366.

(3) With separate control, the number of type 2 connections drops faster than with aggregate control when the number of type 1 connections increases from 0 to 10.

(4) With separate control, the number of type 1 connections does not drop greatly when the number of type 2 connections increases from 0 to 1.

The reason for observation (2) is that the individual CLR for a strong-burst connection is higher than one for a weak-burst connection when using aggregate control, as explained in Sect. 4.2.2. Therefore the number of type 2 connections is limited when there are type 1 connections. Figure 4.15 shows the aggregate CLR and the individual CLRs for traffic types 1, and 2. CLR 1 is clearly the bottleneck in aggregate control.

The reason for observation (3) is as follows. Type 1 connections are more bursty than type 2 ones, so when the number of type 1 connections is small, there is little or no statistical multiplexing gain. Because the sum of the peak rates for type 1 connections is used for separate control, the bandwidth efficiency for type 1 connections is lowered. In contrast, because bandwidth unused by type 1 connections can be used by type 2 ones when using aggregate control, the number of type 2 connections is increased. Figure 4.16 shows the relationship between the bandwidth allocated to type 1 connections and the number of type 1 connections. The allocated bandwidth increases sharply when the number of connections is small. Recall that although we can increase the cell-level performance by using idle slots allocated to other queues, we cannot improve the number of connections, as explained in Section 4.2.1.

The reason for observation (4) is that we do not have to reduce the bandwidth allocated to type 1 connections in order to support a small number of type 2 connections because the peak rate and burstiness factor of type 2 connections are small.

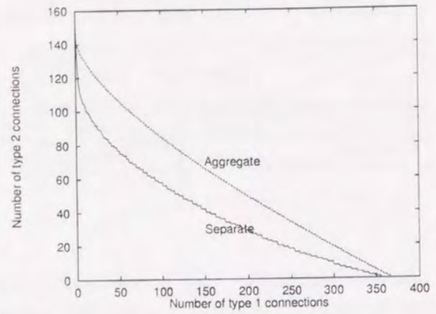


Figure 4.14: Number of type 2 connections for match 1 as a function of the number of type 1 connections.

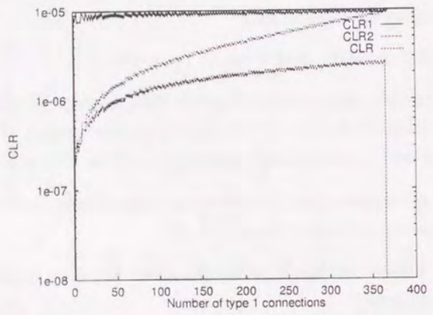


Figure 4.15: CLRs for match 1 when using aggregate control.

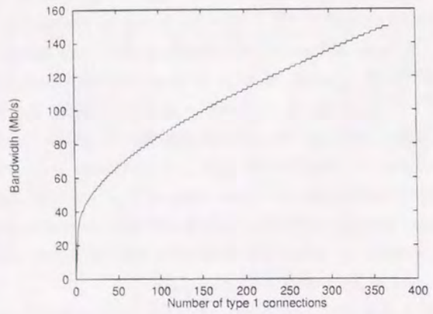


Figure 4.16: Allocated bandwidth for type 1 connections for match 1.

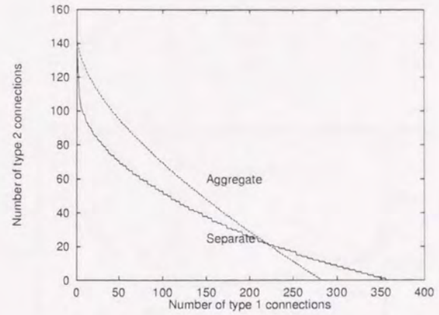


Figure 4.17: Number of type 2 connections for match 2 as a function of the number of type 1 connections.

In Fig. 4.17, a strong-burst type 1 connection is requesting a high CLR while a weak-burst type 2 connection is requesting a low CLR. From this figure, we observe that

(1) Separate control outperforms aggregate control when the number of type 1 connections is above about 220: Otherwise, aggregate control outperforms separate control.

(2) With aggregate control, when the number of type 2 connections increases from 0 to 1, the number of type 1 connections drops sharply from 370 to 281. In contrast, when the number of type 1 connections increases from 0 to 1, the number of connections drops slightly from 142 to 139.

(3) With separate control, when the number of type 2 connections increases from 0 to 1, the number of type 1 connections drops slightly.

(4) With separate control, the difference between the number of type 2 connections for matches 1 and 2 is very small.

The reason for observation (1) is as follows. When the number of type 2 connections is small, only a small bandwidth is needed to satisfy the QoS targets for type 2 connections. In addition, because the bandwidth allocated to type 1 connections is large enough to gain a statistical multiplexing effect, it does not affect type 1 connections to re-allocate a small fraction of the bandwidth to type 2 connections. The reason for observation (2) is that the number of type 1 connections is limited when there are type 2 connections because type 2 connections are requesting a low CLR. Figure 4.18 shows the CLRs with aggregate control. Here, CLR 2 is the bottleneck. The reason for observation (3) is the same as that for observation (3) for match 1. The reason for observation (4) is that the relationship between the number of type 2 connections and the required bandwidth is not sensitive to differences in the CLR. Figure 4.19 shows the relationships among the number of type 2 connections, the required bandwidth, and the target CLR. The relationship between the number of connections and the required bandwidth is not clearly sensitive to the difference between CLRs of 1.0e-5 and 1.0e-7.

In Fig. 4.20, a strong-burst type 1 connection is requesting a low CLR while a weak-burst type 2 connection is requesting a high CLR. From this figure we observe that

(1) Aggregate control outperforms separate control for all the region.

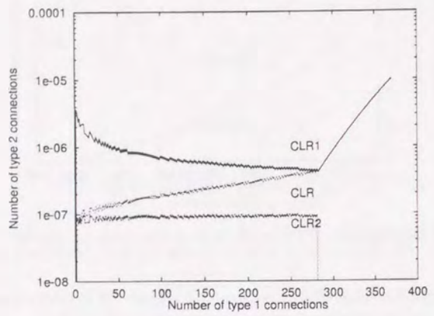


Figure 4.18: CLRs for match 2 with aggregate control.

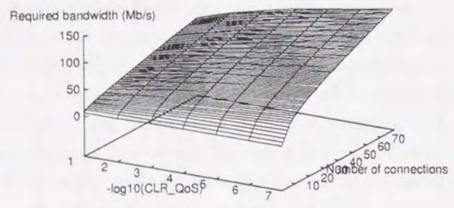


Figure 4.19: Relationships among number of type 2 connections, the required bandwidth, and the target CLR.

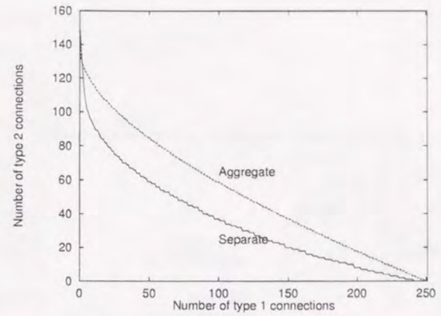


Figure 4.20: Number of type 2 connections for match 3 as a function of the number of type 1 connections.

(2) With aggregate control, when the number of type 1 connections increases from 0 to 1, the number of type 2 connections drops from 152 to 131. In contrast, when the number of type 2 connections increases from 0 to 1, the number of type 1 connections decreases slightly from 251 to 248.

(3) With separate control, the number of type 1 connections decreases more sharply than for match 1 (Fig. 4.14) when the number of type 2 connections increases.

The reason for observation (2) is due to the fact that the number of type 2 connections is limited when there are type 1 connections because the type 1 connections are requesting a low CLR while CLR 1 is higher than CLR 2. The reason for observation (3) is that the statistical multiplexing gain is small when the number of type 1 connections is small because they are requesting a CLR lower than that for match 1 under separate control, which reduces the efficiency of the bandwidth allocated to type 1 connections.

In Fig. 4.21, both a strong-burst type 1 connection and a weak-burst type 2 are requesting the same high QoS level. The difference between this figure and Fig. 4.20 is that the target QoS for this figure is smaller. From this figure we observe that

(1) Aggregate control outperforms separate control for all the region.

(2) Compared with Fig. 4.20, the number of type 2 connections is less with separate control, while there is little difference in the number with aggregate control.

The reason for observation (2) is due to the fact that the number of type 2 connections is limited when there are type 1 connections. That is, if the QoS level for type 1 connections is lower than that for type 2 connections, the QoS level for type 2 connections does not matter because CLR 2 cannot be lower than CLR 1.

From the results for two-type case, we can draw the following conclusions. Aggregate control outperforms separate control except when weak-burst connections are requesting a low CLR and the number of weak-burst connections is small. We can thus maximize the number of connections admitted by alternating separate and aggregate controls adaptively according to the number of connections requesting a low CLR, as shown in Fig. 4.22. A mechanism for maintaining cell order integrity when alternating these controls is shown in Fig. 4.23.

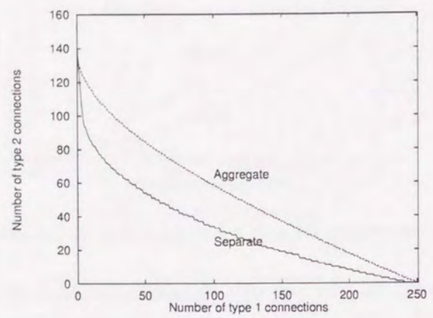


Figure 4.21: Number of type 2 connections for match 4 as a function of the number of type 1 connections.

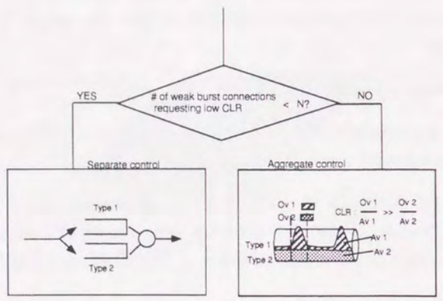


Figure 4.22: Framework for adaptive admission control for multiple QoS requirements.

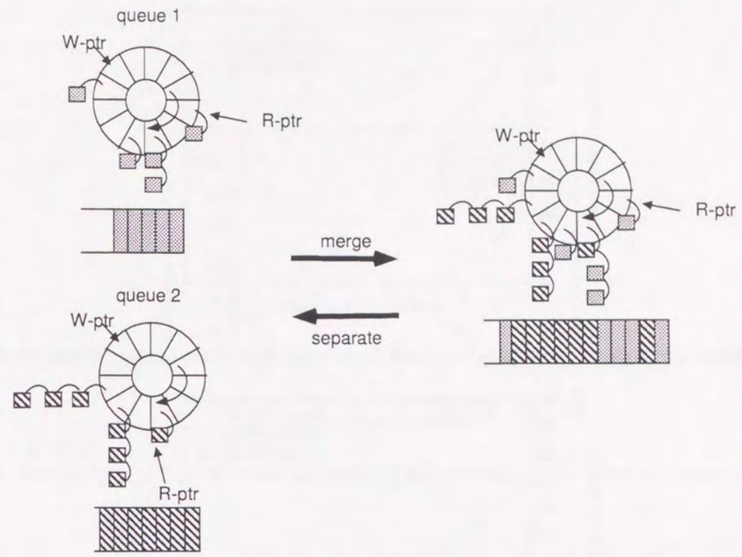


Figure 4.23: Mechanism to maintain cell order integrity when alternating controls.

Table 4.4: Combinations of type and QoS class for three-type case.

Type 1, 2 \ Type 3	High CLR	Low CLR
match 2	match 21	match 22
match 3	match 31	match 32

4.3.2 Three-type case

Next we considered the case in which all traffic three types shown in Table 4.1 coexist, evaluating the number of connections admitted for the combinations of type and QoS class shown in Table 4.4. That is, we evaluated matches 2 and 3 for combinations for type 1 and 2.

We can admit 90 type 3 connections if there are neither type 1 nor 2 connections. We set the number of type 3 connections to 30 or 60 for matches 21, 22, 31, and 32. The number of type 2 connections for matches 21, 22, 31, and 32 as a function of the number of type 1 connections are shown in Figs. 4.24, 4.25, 4.26, and 4.27. From these figures, we observe that aggregate control outperforms separate control for all the region. Based on the results for the two-type case, the difference between aggregate and separate controls is the smallest for match 31 in Fig. 4.26. Aggregate control again still outperforms separate control.

From the results for the two- and three-type cases, we can conclude that except when the number of

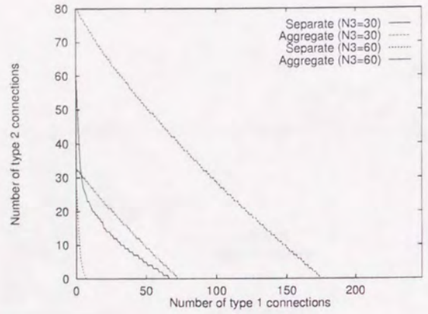


Figure 4.24: Number of type 2 connections for match 21 as a function of the number of type 1 connections.

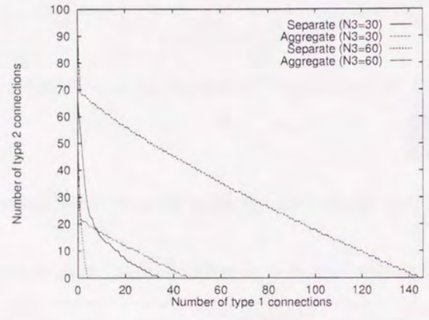


Figure 4.25: Number of type 2 connections for match 22 as a function of the number of type 1 connections.

weak-burst connections requesting a low CLR is small, aggregate control outperforms separate control. The effect of aggregate control is more remarkable when the number of types is larger. By alternating separate and aggregate controls adaptively according to the number of weak-burst connections requesting a low CLR, we can maximize the number of admitted connections. When using separate control, we should minimize the number of queues to avoid segregation loss.

4.4 Concluding remarks

We have developed an efficient and simple admission control for multiple QoS targets. By alternating separate and aggregate control adaptively according to the number of weak-burst connections requesting a low CLR, we can maximize the number of admitted connections. We use a binary-search algorithm to calculate the required bandwidth for queues in separate control. To quickly and simply determine whether a new connection is admitted or not, we reduce the computational complexity in calculating the individual CLRs.

The proposed method is based on the assumption that accurate peak and average rates are given. It is

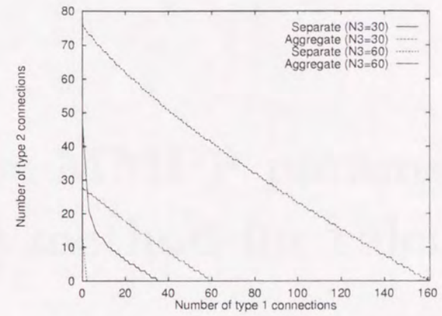


Figure 4.26: Number of type 2 connections for match 31 as a function of the number of type 1 connections.

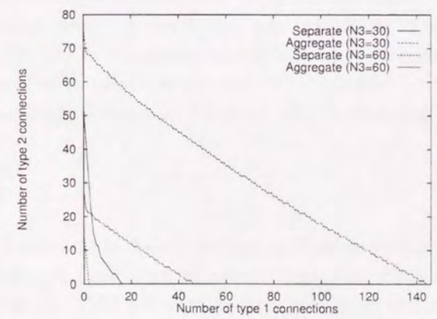


Figure 4.27: Number of type 2 connections for match 32 as a function of the number of type 1 connections.

difficult, however, to declare accurate peak and average rates. Measurement mechanisms for these rates and a dynamic bandwidth negotiation mechanism based on these measurement remain to be developed.

Chapter 5

A real-time MMPP parameter estimating method for calculating cell loss ratio

Since admission control is *preventive* control, we need to establish whether or not an admission control algorithm actually works appropriately. The role of admission control is to maintain an extremely low target cell loss ratio (CLR), e.g., 1.0e-5 to 1.0e-8. This makes it hard to judge whether or not the target CLR level is maintained. As described in this chapter, our approach to tackling this problem was to measure cell streams, fit them to a mathematical model, and solve the queuing systems mathematically. We developed a real-time algorithm for estimating the Markov modulated Poisson process (MMPP) parameters to calculate the CLR. This algorithm employs a fictitious queue and a window to determine the state, i.e., underload or overload. The algorithm's performance is evaluated through computer simulation, and it is shown that the CLR can be appropriately estimated by setting the window size to the minimum cell interval of the connection. It is suggested that the proposed algorithm can be applied to admission control. We conducted computer simulation and confirmed that the admission control method based on this algorithm achieves high bandwidth efficiency.

5.1 Introduction

An attractive feature of ATM networks is their statistical multiplexing ability. By admitting a large number of connections, we can achieve high bandwidth efficiency. High bandwidth efficiency may, however, lead to low Quality of Service (QoS) levels. ATM networks perform rate-based traffic control. Connection admission control (CAC) and usage parameter control (UPC) are key control methods in ATM traffic control. The main objective of CAC is to guarantee, for each service, an appropriate QoS level while enhancing network utilization. CAC decides whether or not a new connection can be admitted by taking into account its impact, which depends on the traffic characteristics of the connection and the QoS levels of existing connections [8, 9]. UPC monitors user cell traffic and checks if it conforms to the traffic descriptor declared at connection setup. Non-conforming cells are discarded or marked by the UPC device. Thus, CAC assumes that only conforming

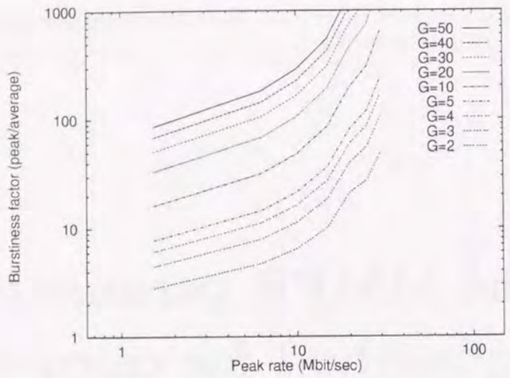


Figure 5.1: Statistical multiplexing gain.

cell traffic may be injected into networks.

Statistical multiplexing gain, G, is defined as the ratio of the maximum number of connections admitted into a link by CAC to the number permitted by the peak assignment strategy. Statistical multiplexing gain depends on the traffic characteristics of the connections, such as peak rate and average rate. We define the connection burstiness factor as the ratio of peak rate to average connection rate. Figure 5.1 shows statistical multiplexing gain, G, as a function of both connection peak rate and connection burstiness factor for maintaining a CLR (Cell Loss Ratio) of 10^{-6}. In Fig. 5.1, a link capacity of 149.76 Mbit/sec is assumed, and CLR is evaluated by using virtual cell loss probability in the bufferless fluid flow model [52]. Virtual cell loss probability is defined as the ratio of overflow cell rate to offered cell rate [52]. No connection traffic characteristics are assumed, other than connection peak and average rate in the virtual cell loss probability. According to Fig. 5.1, we can achieve a statistical multiplexing gain, G, of up to 50 for traffic with a connection peak rate of 10 Mbit/sec, and a connection burstiness of 100. This means that up to 50 times the number of connections can be accommodated by statistical multiplexing, and that the bandwidth cost can thus be reduced by up to 98 (=100 $\times$ (1-$\frac{1}{50}$))%.

As shown above, we can achieve high bandwidth efficiency while maintaining the desired QoS level by using appropriate CAC methods. CAC and UPC are preventive control methods. If they malfunction, QoS may degrade. For example, if UPC malfunctions, non-conforming cells may be injected into networks, thus leading to congestion. If CAC malfunctions and is allocated a bandwidth smaller than requested by the connection, there may not be enough bandwidth to maintain the QoS levels. We need to monitor the QoS levels provided by these controls and check if the target levels are maintained.

In this chapter, we consider the mechanism for monitoring QoS levels. CLR is a QoS measure considered in designing CAC and UPC. The target CLR is considered to be very low, e.g., 1.0e-5–1.0e-8. It takes a considerable time to accurately measure the CLR ranging over such low values. Instead of measuring the CLR directly, we measure a cell stream, fit it to the mathematical model, and calculate the CLR by solving the model.

Since ATM networks handle multimedia traffic, cell traffic characteristics are bursty. Several bursty traffic models have been constructed. The Markov modulated Poisson process (MMPP) is one example. MMPP is a

doubly stochastic Poisson process, in which several states are modulated according to the Markov process. In each state, cells are generated according to a Poisson process whose arrival rate is intrinsic to the state. Queuing systems loaded by MMPP input have been analyzed by many researchers [22, 23, 24, 25]. Two-state MMPP is shown to be efficient for modeling multiplexed bursty traffic in high-speed networks [51]. Two-state MMPP is a parsimonious model: it is characterized by only four parameters, two arrival rates and two transition rates between states. Therefore, we adopt MMPP as a model to which a multiplexed cell stream is fitted.

In Sect. 5.2, we develop a parameter estimating method, in which a multiplexed cell stream is fitted to two-state MMPP. This method assumes that the cell stream is modeled by two-state MMPP composed of overload and underload states. The developed algorithm employs a fictitious queue and a window to determine the load states, i.e., underload or overload. The performance of this method is investigated through simulation for multiplexing of on-off sources with a wide spectrum of traffic characteristics. It is shown that the CLR can be appropriately estimated by setting the window size to the minimum cell interval of the connections. Sect. 5.3 proposes an improvement for the proposed method. The improved method can be applied to both homogeneous and heterogeneous connections with respect to peak rate. In Sect. 5.4, it is suggested that the proposed algorithm can be applied to admission control. Through computer simulation, we show that this admission control method achieves high bandwidth efficiency. Sect. 5.5 evaluates the admission control algorithm for both homogeneous and heterogeneous cases. Sect. 5.6 summarizes this chapter. Since we measure cell stream and employ two-state MMPP, which simplifies the measuring algorithm, we can accurately and quickly estimate the CLR provided by traffic control methods used in the ATM network.

5.2 Parameter estimating method for multiplexed cell stream

Two-state MMPP is used to model multiplexed cell streams; each stream is on-off source [33, 51]. Here we review these methods and propose our parameter estimating algorithm.

5.2.1 How to fit multiplexed cell streams to MMPP

Moment matching method

Heffes et al. developed a method for estimating a two-state MMPP model from actual voice traffic data [33]. They estimated MMPP parameters by matching average, variance, and the third central moment of the counting process $N(t)$, where $N(t)$ is the number of cells observed in a period of $[0, t]$. Let λ_1 and λ_2 denote cell arrival rate at high and low state. Let r_1 and r_2 denote transition rate from low to high state and from high to low state. Define $b_t = var[N(t)]/E[N(t)]$ and $b_\infty = \lim_{t\to\infty} var[N(t)]/E[N(t)]$. Let d be a solution of the following equation.

$$d = \frac{1}{t_1}\left(\frac{b_\infty - 1}{b_\infty - b_{t_1}}\right)(1 - \exp(-dt_1)). \tag{5.1}$$

Let

$$e = \frac{(b_\infty - 1)ad^3}{2K^2},$$

we have

$$r_1 = \frac{d}{2}\left(1 + \frac{1}{\sqrt{4e+1}}\right), \tag{5.2}$$

$$r_2 = d - r_1, \tag{5.3}$$

$$\lambda_1 = \frac{K}{r_1 - r_2} + \lambda_2, \tag{5.4}$$

$$\lambda_2 = \left(\frac{ad}{r_2} - \frac{K}{r_1 - r_2}\right)\left(\frac{r_2}{r_1 + r_2}\right), \tag{5.5}$$

where $a = E[N(t)]/t$ and K is given by

$$K = \frac{d^3(g^3(1,t_2) - a^3t_2^3 - 3a^2t_2^2(b_\infty - 1)) + 3a^2d^2t_2(b_\infty - 1)(1 - \exp(-dt_2))}{3a(b_\infty - 1)(dt_2(1 + \exp(-dt_2)) - 2(1 - \exp(-dt_2)))}, \tag{5.6}$$

where

$$g^3(1,t_2) = \mu_3^*(t_2) + 3at_2(at_2 - 1)b_{t_2} + at_2(at_2 - 1)(at_2 - 2), \tag{5.7}$$

where $\mu_3^*(t_2)$ is the third central moment of the counting process.

$$\mu_3^*(t) = E[(N(t) - E[N(t)])^3]. \tag{5.8}$$

This method is too complicated to perform in real-time estimation because the computation includes evaluation of an exponential function in Eq. (5.6) and we need to solve Eq. (5.1). In addition, we need to choose t_1 and t_2 so that $Var[N(t)]$ has a good fit to that of MMPP. Thus, we cannot take this approach to estimating real-time parameters as it is.

Asymptotic matching method

Baiocchi et al. developed a method for estimating parameters of two-state MMPP by dividing overload and underload states [51]. Suppose that N on-off sources are multiplexed and that M is the maximum number of active connections to ensure that overload does not occur. Parameters for two-state MMPP are given as follows.

$$r_1 = \eta, \tag{5.9}$$

$$r_0 = r_1\frac{Np\Lambda - \lambda_0}{\lambda_0 - Np\Lambda}, \tag{5.10}$$

$$\lambda_1 = \Lambda\sum_{i=M+1}^{N} i\frac{\pi_i}{\sum_{j=M+1}^{N}\pi_j}, \tag{5.11}$$

$$\lambda_0 = \Lambda\sum_{i=0}^{M} i\frac{\pi_i}{\sum_{j=0}^{M}\pi_j}, \tag{5.12}$$

where $-\eta$ is a maximal real part eigenvalue of an infinitesimal generator for a transient Markov process, obtained from the phase process considering only the states $\{M, M+1, \ldots, N\}$. Λ is the cell emission rate from an active source. p is a inverse of burstiness factor.

The asymptotic matching method gives a physical meaning to the fitting process. The overload state is defined as the state in which the total cell rate exceeds the link capacity, while the underload state is defined as the state in which the total cell rate does not exceed the link capacity. This method estimates the accurate CLR of a statistical multiplexer loaded with homogeneous on-off sources [51]. A set of traffic descriptors for all the connections is needed to calculate the parameters by the asymptotic matching method. For real-time estimation of MMPP parameters we need to measure actual traffic instead of using a traffic descriptor.

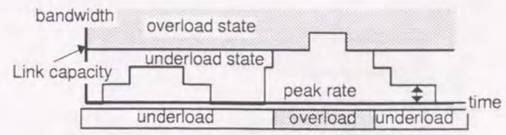


Figure 5.2: Two-state model.

5.2.2 Measurement-based asymptotic matching method

We propose a real-time MMPP parameter estimating method called *measurement-based asymptotic matching*. The idea behind this method is similar to that behind the asymptotic matching method, except that our method is based on measurement. The success of asymptotic matching is due to the fact that the occurrence of the overload condition causes most of the cell losses when on-off sources are multiplexed. The cell stream is modeled by overload state and underload state as illustrated in Fig. 5.2. The overload state is defined as the state where the cell arrival rate from all the connections exceeds the link capacity. The underload state is defined as the complement of the overload state.

Thus, we measure the number of cells within a measurement window and detect the overload situation. The measurement algorithm is described as follows.

Step 1: Count the number of cells within a measurement window, w.

Step 2: Compare the number of cells with a threshold, X. If it is larger than X, we determine a high state. If it is smaller, we determine a low state.

Step 3: According to the result of Step 2, we modify the measurement data of the corresponding state.

Appropriate determination of the threshold value, X, and window size, w, is critical. They must be determined such that $X/w = 1$ holds. That is, we can determine X once we have w. In the following section, we evaluate the relationship between accuracy and window size, w.

5.2.3 Accuracy of estimated parameters and CLR

This section evaluates the accuracy of our proposed method. We assume that the connection peak rate $R =$ 10 Mbit/sec, burstiness factor defined as the ratio of peak to average rates $a = 10$, and mean burst size $B =$ 16250 Bytes. We estimate the parameters for the offered load ranging from 0.3 to 0.8 by changing the number of connections.

Figures. 5.3 and 5.4 show the relationship between arrival rates of high and low states and the offered load, and that between the mean durations of high and low states and the offered load. We also plot curves for parameters obtained by the asymptotic matching method. From Fig. 5.3, it is observed that the estimated parameters agree well with those obtained by the asymptotic matching method for all the measurement window size, w. However, there are deviations between the estimated parameters and those obtained by the asymptotic matching method for window size, $w = 10$ cells. The window size of 10 cells underestimates the mean duration

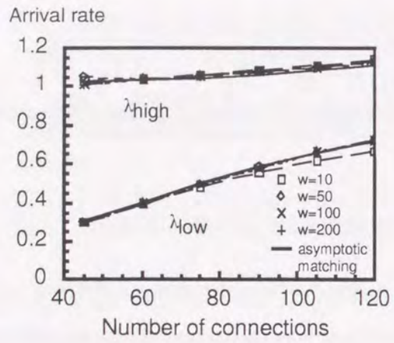


Figure 5.3: Relationship between arrival rates of high and low states and the offered load.

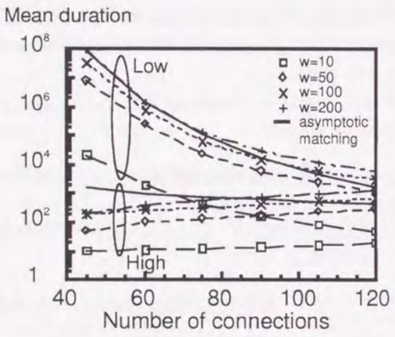


Figure 5.4: Relationship between mean durations of high and low states and the offered load.

time. This is explained as follows. Each connection sends cells at a peak rate of 10 Mbit/sec, resulting in a minimum cell interval of 15 cells. As long as the connection stays in the on state, it sends cells continuously at a cell interval of 15 cells. If the measurement window size is larger than the minimum cell interval, at least one cell over each connection is observed within a measurement window. However, if the window size is smaller than the minimum cell interval, one cell over each connection is not necessarily observed within a measurement window, leading to misdetection of the end of the overload state. Thus, when the measurement window is smaller than the minimum cell interval, the estimated mean duration is smaller than the actual value.

By solving the MMPP/D/1/K queuing model, we calculated the CLR for the estimated MMPP model. We assumed the buffer size to be 20 cells, and used the analysis presented in [23]. Figure 5.5 shows the relationship between the CLR and the number of connections. We also plotted the CLR obtained by computer simulation. From this figure, we observe that the window size of 10 cells underestimates the CLR by an order of magnitude, while window sizes of other than 10 cells estimates the CLR accurately.

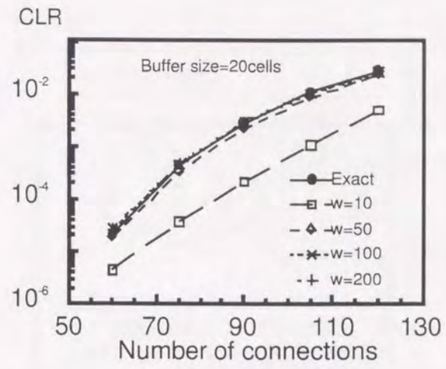


Figure 5.5: Relationship between CLR and number of connections.

Next, we evaluate the impact of peak rate, burstiness factor, and mean burst length. Figures 5.6, 5.7, and 5.8 show the relationship between CLR and peak rate, burstiness factor, and mean burst length. For the peak rate (Fig. 5.6), we observe that the proposed method estimates conservative and accurate CLRs for peak rates over the range of practical interest. For the burstiness factor (Fig. 5.7), we observe that the proposed method estimates conservative and accurate CLRs for burstiness factors larger than 10. The proposed method achieves a conservative CLR for burstiness factors smaller than 10, even though the deviations are larger than those for burstiness factors larger than 10. Since we cannot expect high statistical multiplexing gain within the region in which the burstiness factor is smaller than 10 (recall Fig. 5.1), the deviation observed in the region of burstiness factors smaller than 10 cannot be a major drawback. For the mean burst length, we observe that the proposed method estimates conservative and accurate CLRs for all the values used.

5.3 Improved measurement-based asymptotic matching method

Window sizes larger than the minimum cell interval yield accurate parameters and CLRs because at least one cell over each connection is observed within a measurement window whose size is larger than the minimum cell interval of the connection. If we assume homogeneous connections, there is no problem with a measurement window whose size is the minimum cell interval of the connection. For heterogeneous connections, we need to modify the measurement algorithm for the following reason.

In the previous algorithm in Sect. 5.2.2, we used the nominal rate defied as X/w. If we assume homogeneous cases, when the state becomes high, then X/w is made larger than or equal to one by setting the w to the minimum cell interval of the connection (Note that X is set equal to w, as mentioned in Sect. 5.2.2). If we use X larger than the minimum cell interval, X/w may become smaller than one even if the state becomes high. For heterogeneous connections with respect to peak rate, we cannot determine one value for w. Thus, we need to develop a mechanism for detecting a high state while allowing connections with different peak rates to exist.

We propose an improved measurement-based asymptotic matching method, in which a fictitious queue and

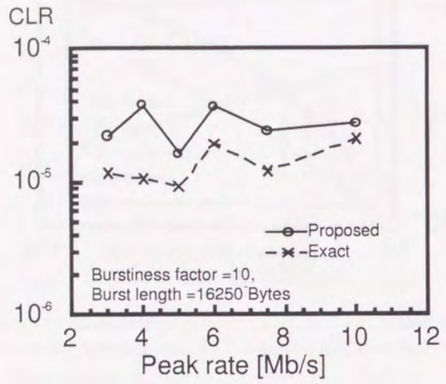


Figure 5.6: Relationship between CLR and peak rate.

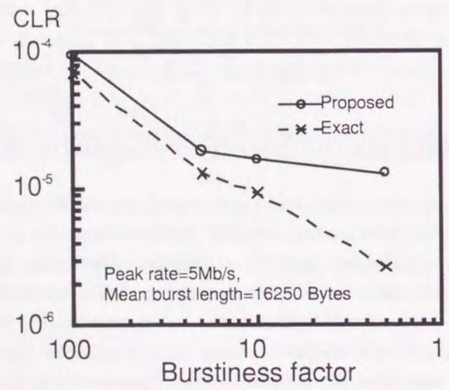


Figure 5.7: Relationship between CLR and burstiness factor.

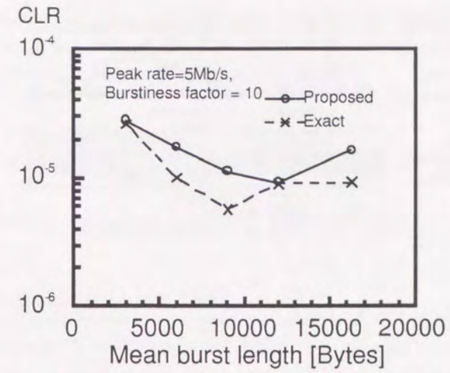


Figure 5.8: Relationship between CLR and average burst length

a window are employed to detect the overload state. Figure 5.9 shows the mechanism for detecting the overload state. A *fictitious queue* and a window are used to detect the overload state. Overload state is detected when fictitious queue has not been empty for a certain period (we call it a *window*). The details are explained in the following. We assume the time axis is slotted by a unit of time δ during which one cell is sent on to a link. This unit of time δ corresponds to about 2.7 μ sec on a 150 Mb/s link.

- The length of fictitious queue is incremented on a cell arrival. We assume cell arrival occurs at the beginning of every slot. Thus, on arrival of cells, fictitious queue is incremented by the number of cells.
- Decreasing rate of fictitious queue length is equivalent to the link transmission rate. The length of fictitious queue is decremented at the end of every slot if fictitious queue is not empty.
- Whether or not the overload state starts is determined at the beginning of each slot during underload states. The overload state is detected when fictitious queue has not been empty at the beginning of consecutive slots over a certain period (we call it *window size*). A window is introduced to eliminate the mis-detection of start and/or end of the overload state. While a too small window leads to mistaking the underload state for the overload state, a too large window leads to missing the overload state. The optimum window size is examined in Sect. 5.5.
- Whether or not the overload state ends is determined at the beginning of each slot during overload states. After the overload state ends, fictitious queue monotonically decreases and eventually becomes empty. The start time of the monotonical decrease of fictitious queue length is regarded as the end time of the overload state.

If no cell arrives and the fictitious queue is empty, underload is detected. On arrival of cells, at the beginning of the slot, fictitious queue is incremented by the number of cells. At the slot, the state is not changed, but it becomes a candidate for the overload state.

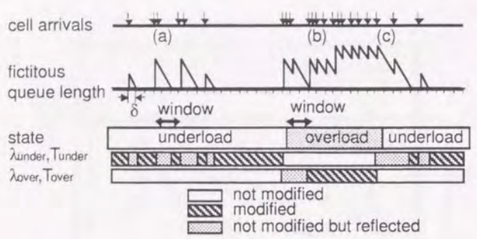


Figure 5.9: Mechanism of detecting overload state.

If fictitious queue has not been empty over a window size, it is regarded as the overload state. In Fig. 5.9 we assume the window size of 3δ. The period during which fictitious queue is not empty but becomes empty after 2δ is regarded as the underload state as shown in (a), but the period during which fictitious queue is not empty over 3δ from its start time is regarded as the overload state as shown in (b). We can not decide whether it is the overload state or the underload state during the window size slots from when fictitious queue becomes non-zero. The parameters for each state are not modified during the period. The statistics gathered during the period are reflected in the parameters for the overload state: λ_{over}, T_{over}, after the window size. If the overload ends, fictitious queue begins to decrease. We define the end of the overload state when fictitious queue is a maximum during the window from the time shown in (c) of Fig. 5.9. We can not decide the end of the overload state during the window size slots from when fictitious queue becomes a maximum. The statistics gathered during the period are reflected in the parameter for the underload state: λ_{under}, T_{under}.

So far we have described the method for detecting the overload state. In the following part of this section, we present the method for estimating the parameters for the overload and underload states. Theoretically, the cell arrival rate of the overload state is obtained by dividing the *total* number of cells arriving during the overload state by the *total* duration of the overload state. Since this estimation equally weights all data, it tends to neglect the information on current trends as time elapses. In addition, because all cell counters have finite capacity, this method can not be implemented. To overcome this problem, we employ a first-order low-pass filter. As an example, the cell arrival rate of overload state is derived as follows. (Note that the underload state case can be derived in a corresponding manner).

We apply the first-order low-pass filter to the arrival rate and the mean sojourn time of the overload state using the smoothing coefficients α and β. We assume that the overload state starts at time τ_h and continues to time t ($t \geq \tau_h + w$), and that the number of cells arriving at time t is $a(t)$. The observed cell arrival rate $\lambda_{over}(t)$ and the observed mean sojourn time $T_{over}(t)$ are evaluated as follows :

$$\lambda_{over}(t) = \alpha a(t) + (1-\alpha)\lambda_{over}(t-\delta) \quad (5.13)$$

$$T_{over}(t) = \beta(t-\tau_h) + (1-\beta)T_{over}(\tau_h) \quad (5.14)$$

where $0 \leq \alpha, \beta \leq 1$. The observed cell arrival rate and the observed mean sojourn time of the underload state are evaluated in the same manner. The coefficient α in equation (5.13) controls the cut-off fluctuation frequency

of the cell arrival rate during one slot. Small α eliminates the effect of the high frequency component of the cell arrival rate. β in equation (5.14) controls the fluctuation of the mean sojourn time of the overload state. This chapter focuses on the effect of α, which is investigated later.

5.4 Application to admission control

Here we describe application of our estimating method to admission control.

5.4.1 Framework for guaranteeing QoS

Figure 5.10 illustrates the framework of the proposed adaptive admission control. First, the currently used bandwidth is monitored using measurements of cell stream, and the bandwidth utilization is updated (STEP 1). If a new connection setup request exists, the nominal bandwidth utilization after accepting the connection is anticipated by taking into account the traffic descriptor (STEP 2). The declared peak rate of the connection is used for this purpose. Then the CLR is evaluated (STEP 3). If the evaluated CLR is lower than the QoS objective, the connection setup request is accepted and the bandwidth utilization is substituted by the nominal one obtained in STEP2 (STEP 4). We need to evaluate the CLR using the bandwidth utilization in STEP 3. The MMPP parameter estimating method is used. CLR can be evaluated by solving the MMPP/D/1/K queuing model [23].

5.4.2 Admission of a new connection

If a new connection setup request is issued, the nominal bandwidth utilization after its setup should be anticipated by taking into account the traffic descriptor of the requested connection.

Peak rate, average rate and mean burst length are considered as important traffic characteristics. While peak rate can be estimated from the performance of terminals and/or UNI interface speed beforehand, other characteristics are hard to anticipate. In many cases, the peak rate assignment strategy leads to inefficient network utilization. With the proposed method, even if the nominal bandwidth utilization includes some margin, it will converge to the actual value through repeated measurement. Thus, we adopt only peak rate as a traffic descriptor of a new connection since it will give a conservative estimate. Let the peak rate of a new connection be R and the link capacity be C. Thus, we calculate the nominal cell arrival rates of each state as follows :

$$\lambda_{over}{}^{+} = \lambda_{over} + R/C$$

$$\lambda_{under}{}^{+} = \lambda_{under} + R/C$$

Mean sojourn times are not modified.

If a new call is bursty, these estimates contain safety margins. They approach the value for the actual traffic with time.

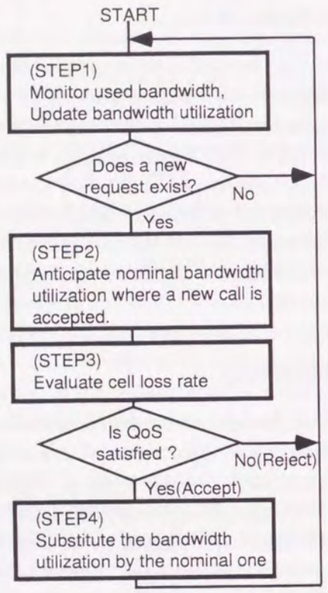


Figure 5.10: Framework of adaptive admission control.

5.5 Performance evaluation

Next, we investigate the bandwidth efficiency achieved by our method. The performance measure is the bandwidth efficiency when the maximum number of connections, QoS of which are satisfied, are admitted.

5.5.1 Evaluation model

We assume that connections are multiplexed onto a link whose capacity is C (Mb/s). The performance of the multiplexer loaded with on–off sources is evaluated, where connections are characterized by heterogeneous on–off sources.

An on–off source is often used as a model of data traffic in ATM networks[51]. The source continuously emits burst data, which is segmented into cells at the peak rate, during the on state (the source in the on state is called *active* source below). The off state corresponds to the silent periods between data bursts. We assume the duration of both on and off periods are exponentially distributed.

We introduce some notations for type i connection:

- N_i : number of connections,
- B_i : mean burst length (Byte),
- R_i : peak rate (Mb/s),
- a_i : burstiness factor (= peak rate/average rate).

The following evaluation assumes that the link capacity is 150Mb/s, the buffer capacity is 20 cells and the QoS objective is 10^{-5}. Throughput deterioration due to the go-back-N retransmission mechanism is negligible when cell loss rate is 10^{-5} [46].

5.5.2 Optimum window size

Figure 5.11 shows the relationship between the window size and the bandwidth efficiency achieved by our method, and the window size for three mixing patterns of two types of connections (confidence intervals are within a few percentiles); the ratios between lightly loaded connections (1.5 Mb/s) and heavily loaded ones (6 Mb/s) are 10:0, 8:2, and 4:6.

Figure 5.11 also shows two extreme admission controls. One is optimum control using the average rate and mean burst length of each call. If an average rate and a mean burst length is obtained beforehand, we could admit the maximum number connections. An average rate and a mean burst length are, however, hard to predict at connection set-up, so we cannot adopt this method. The other extreme one is peak assignment control. Since only peak rate is used in this control, it can not achieve high bandwidth efficiency for bursty traffic. Since we assume $a_i = 10$, the peak assignment control achieves only 0.1 bandwidth efficiency.

Figure 5.11 shows that our method observes the QoS objective, since the bandwidth efficiency it achieves is lower than that achieved by optimum control. Figure 5.11 shows that the optimum window size is the peak cell interval (C/R_1=25) for the homogeneous case (1). A window smaller than 25 cells does not enhance bandwidth efficiency. However, a window size of w over 25 cells achieves nearly optimum bandwidth efficiency. This is explained below.

Small windows are very sensitive to cell arrival since they indicate the start of an overload state when even a few cells have arrived within a short period. Therefore, small windows result in accurate cell loss ratio estimation, and so do not significantly improve resource utilization. Conversely, windows larger than peak cell

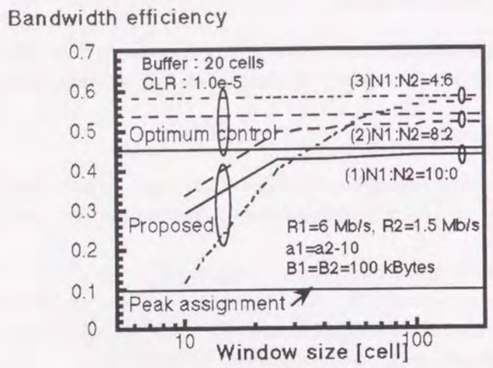


Figure 5.11: Relationship between window size and bandwidth efficiency.

intervals more accurately detect the overload state where the cell emission rate from all connections exceeds link capacity. If the cell emission rate from all the connections exceeds the link rate, the fictitious queue length never becomes zero in a peak cell interval. Thus, if the window size is set to the peak cell interval for the homogeneous case, the method never takes any underload state for the overload state. A window size larger than the peak cell interval tends to miss overload states whose durations are shorter than the window size. In addition, a long window is not suitable for real-time control. A window size equal to the peak cell interval is optimum for the homogeneous case.

Figure 5.11 shows that as the ratio of low speed connections increases, w must increase and approach the peak cell interval of low speed connection (C/R_2=100). A window size of 100 cells nearly achieves the optimum bandwidth efficiency for all possible connection ratios. In this case, the behavior of a connection whose peak rate is 1.5 Mb/s is not negligible for a link whose capacity is 150 Mb/s. However, the behavior of a connection whose peak rate is lower than one percent of the link rate does not have any impact on QoS [74]. In a heterogeneous environment, we should select the window size of the peak cell interval of the lowest rate type connection which has an impact on QoS.

For statistical multiplexing gain, our method achieves bandwidth efficiencies three to five times higher than those offered by the peak assignment policy. Although for the sake of convergence of simulation runs, we used a burstiness factor a of 10, the method yields even more benefits with very bursty traffic (say, a =100).

5.5.3 Impact of traffic characteristics on performance

The applicability of our method was examined for a wide variety of input traffic parameters.

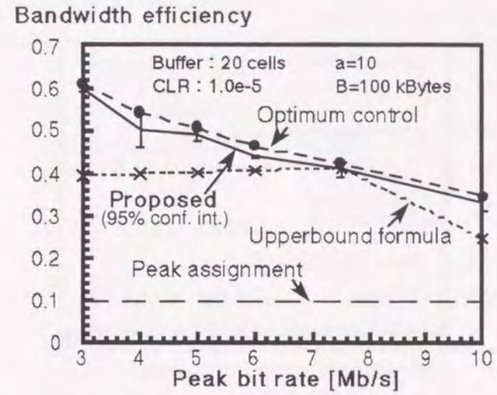


Figure 5.12: Relationship between peak rate and bandwidth efficiency.

Peak rate

Figure 5.12 shows the relationship between peak rate and bandwidth efficiency for homogeneous connections with peak rates R ranging from 3 to 10 Mb/s. Bandwidth efficiency achieved by the proposed method is plotted with 95% confidence intervals. We assume a=10, B=100 kByte. The window size is assumed to be the optimum value for each peak rate (namely w= C/R). Figure 5.12 also shows the bandwidth efficiency achieved by an upper bound formula [11].

The proposed method nearly achieves optimum bandwidth efficiency for the whole range of peak rate. The upper bound formula is based on the cell arrival distribution which does not include any other assumption. This non-parametric nature of the upper bound formula is an attractive feature, but it is hard to approach optimum bandwidth efficiency.

As the peak rate becomes larger, the bandwidth efficiency falls. This is because the connection behavior starts to impact QoS. At a peak rate of over 10 Mb/s, we cannot generally expect significant statistical multiplexing gains [74]. The applicability of our method is supported by the fact that it achieves close to optimum bandwidth efficiency at peak rate at 10 Mb/s for a link capacity of 150 Mb/s.

Mean burst length

Figure 5.13 shows the relationship between mean burst length and bandwidth efficiency. Bandwidth efficiency achieved by the proposed method is plotted with 95% confidence intervals. The homogeneous case is evaluated. Mean burst length B ranges from 1 to 100 kBytes, and we assume R=6 Mb/s, a=10. Window size is assumed to be the optimum value of 25 (= C/R) cells.

Figure 5.13 shows that the bandwidth efficiency of our method is slightly lower than that of optimum control for short burst length. As burst length increases, our method achieves nearly optimum bandwidth efficiency. The reason for this is as follows.

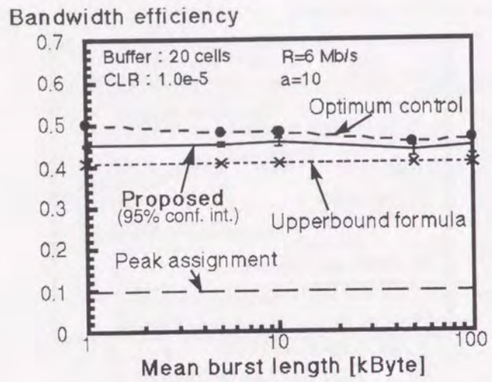


Figure 5.13: Relationship between mean burst length and bandwidth efficiency.

A buffer can absorb the burst level contention for a short burst. The randomness of cell level behavior is a dominant factor in cell loss for a short burst. As burst length increases, a buffer of feasible size (say, 10–1000 cells) cannot absorb the burst level contention. Thus, burst level behavior is a dominant factor in cell loss for a long burst.

The proposed method, however, emphasizes the burst level behavior by detecting the overload state wherein the total emission rate from all the active connections exceeds the link capacity. That is, the method does not accurately evaluate the cell loss rate for short bursts. It does, however, achieve reasonable bandwidth efficiency for all burst lengths evaluated here.

5.5.4 Evaluation of coefficient α

Next, we examine the performance of our method for the non-stationary case, where arrivals and departures of connection occur.

If a new connection is admitted, cell arrival rates are modified using peak rate so that they include a safety margin. These rates are updated using measurements (see Eq. (5.13)). The safety margins decrease as measurement proceeds. Coefficient α in Eq. (5.13) determines the period in which these rates approach the value for the actual traffic. We study the saturated case, where the number of connections does not decrease.

We assume that the interval of connection request arrivals is exponentially distributed with a mean of about 1.4 sec and that each connection is not completed. The number of connections monotonically increases until a setup request is rejected. Each connection has $R = 6$ Mb/s, $a = 10$, $B = 10$ kBytes. The QoS objective of cell loss rate is assumed to be 10^{-5}. From a simulation study, we determine that a maximum of 116 connections is required to maintain the QoS objective for a link whose capacity is 150 Mb/s .

Figure 5.14 shows a sample path of the number of accepted connections. This is obtained from a single run of a computer simulation. Coefficient α ranges from 10^{-2} to 10^{-6} and coefficient β is 10^{-1}. Since the mean

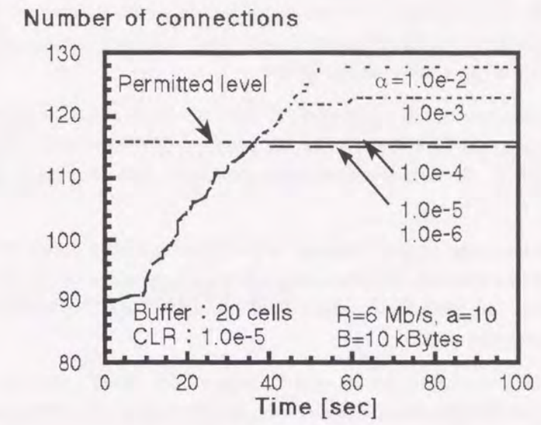


Figure 5.14: Effects of α on the maximum number of admitted connections.

sojourn times are less frequently updated than the arrival rates, the coefficient β should be larger than α to accelerate the convergence of the mean sojourn times.

90 connections are set up at time 0 sec. As connection setup requests arrive, they are always accepted until about 35 sec for all α. After about 35 sec, they are still accepted for over 116 connections for α=10^{-2}, 10^{-3}. However, if more than 116 connections are admitted, we can not keep 10^{-5} of cell loss rate as a QoS objective. Since large α can not eliminate high frequency components of cell arrival rates, it results in mis-admission due to high variation of anticipated cell loss rate.

Conversely, small α eliminates the high frequency components of cell arrival rates. Then, small α produces a stable cell loss rate, which successfully regulates the connection admission. In Fig. 5.14, the connection setup requests over 116 are rejected for α of 10^{-4}. Connection setup requests over 115 are rejected for α from 10^{-5} to 10^{-6}. These values for α regulate the admission of connections well. Thus, we can observe that the best choice of α is 10^{-5} in this case, from the viewpoint of maintaining the QoS objective.

5.6 Concluding remarks

We developed a real-time algorithm for estimating Markov modulated Poisson process (MMPP) parameters so as to calculate CLRs ranging over very low values, e.g., 1.0e-5 to 1.0e-8. The developed algorithm employs a fictitious queue and a window to determine the load states, i.e., underload or overload. The algorithm's performance was evaluated through computer simulation. We showed that the CLR can be appropriately estimated by setting the window size to the minimum cell interval of the connections. We evaluated the effect of the connection peak rate, burstiness factor, and mean burst length on the basis of the accuracy of the estimated CLR. It was thus shown that our proposed method achieves conservative and accurate CLRs for all the values we are interested in.

It was suggested that the proposed algorithm can be applied to admission control. We conducted a computer simulation, which confirmed that the admission control method based on this algorithm achieves high bandwidth efficiency. The followings conclusions were made:

- The method achieves nearly optimum bandwidth efficiency for on-off sources with practical traffic characteristics, and it guarantees the QoS objective for the practical parameter sets used herein (the peak rate ranges from 3 to 10 Mb/s). Statistical multiplexing gains are 3 to 5 for bursty traffic whose burstiness factor is 10.

- The optimum window size is the peak cell interval of the lowest rate type connection which has an impact on QoS. The optimum window size achieves nearly optimum bandwidth efficiency. We used an optimum window size between 25 and 100 cells for mixtures of 1.5 Mb/s and 6 Mb/s connections in a 150 Mb/s link according to the mixture ratio.

Furthermore, we studied the performance for the non-stationary case, where arrivals of connection occur. It was thus shown that if the coefficient α of a first-order low-pass filter applied to cell arrival rate in our method is decided correctly, it protects mis-admission where too many admitted connections results in violation of QoS objectives.

Since we measure the cell stream instead of measuring cell losses directly, we can accurately and quickly estimate the CLR ranging over very low values. Since the method characterizes the cell stream by overload state and underload state, the measurement algorithm is simplified and is suitable for online processing.

Chapter 6

Concluding remarks

The telecommunications industry is changing rapidly. Many multimedia applications are being developed, and traffic volume is growing at an exponential rate as seen in the recent explosive growth of the Internet. To handle this rapid growth and change, future networks must be scalable and adaptive. In this dissertation we presented a scalable ATM switching architecture and an adaptive admission control mechanism. In this architecture, it is easy to scale the switch size, making it adaptable to changes of traffic characteristics. The main theme of our work was to achieve scalability and adaptability of ATM networks. To accomplish these two targets, we developed (1) a scalable ATM switch architecture, (2) a measurement-based adaptive admission control for traffic whose characteristic is unpredictable, (3) an adaptive admission control for multiple QoS levels, and (4) a real-time algorithm for estimating the CLR.

In Chapter 2, we presented an scalable ATM switch architecture, which is based on a cell-bypass queueing discipline. We investigated the effect of this cell bypass queueing discipline for the 2×2 switching elements in a Banyan network. We started by doing an exact analysis of the 2×2 switching elements to derive the mean queue length at input port buffer, the throughput, and the cell blocking probabilities. We then developed an approximate analysis for the entire Banyan network by using our exact analysis. Simulation was used in those cases where analysis could not be applied. Our results showed that a Banyan network using cell bypass queueing discipline has a higher maximum throughput and a lower mean delay time than one using the FIFO discipline.

Application of this analysis method to nonuniform input traffic models showed that there is a significant difference between the mean cell delay time experienced by the cell of the light traffic and those of heavy traffic in the switching element in some traffic situations. This difference is larger when using the cell bypass queueing discipline than when using the FIFO discipline under extremely unbalanced traffic conditions. We also examined this difference for the entire Banyan network, and found that the mean cell delay time does not increase larger with a wider scanning range even in the extremely unbalanced case.

In Chapter 3, we presented our measurement-based adaptive admission control. We presented our method for for designing a low-pass filter that calculates the instantaneous rate and our framework for bandwidth management based on this rate. A recursive type LPF was shown to perform stably regardless of the connection's peak rate. Our bandwidth management framework is based on the maximum instantaneous rate observed during the monitoring period.

Analysis of the distribution of the underload periods, in which the instantaneous rate does not exceed the link capacity, showed that the monitoring period is not affected by the peak rate and the burstiness factor. Analysis of the distribution of the number of admitted connections showed that as the mean holding time increases, the average number of admitted connections increases and becomes saturated. Our proposed method thus regulates the number of connections properly. We compared the performance of our proposed method to that of conventional CAC methods and found that it achieves a higher maximum number of connections. We investigated the impact of long range dependent traffic on the performance of our proposed method and found that it does not degrade the performance.

We demonstrated a simple hardware implementation by dividing the monitoring period into several bins and expressing the smoothing coefficient of the recursive LPF by a power of two. We showed that the computational complexity of our proposed method is low and constant irrespective of the size of the switching fabric.

Because our bandwidth management method does not assume any mathematical model and can be implemented in simple hardware, it is well suited for practical ATM switching systems.

In Chapter 4 we presented our admission control for multiple QoS targets. By alternating separate and aggregate control adaptively according to the number of weak-burst connections requesting a low CLR, we can maximize the number of admitted connections. We use a binary-search algorithm to calculate the required bandwidth for each queue to speed up the bandwidth re-allocation process. We use two simple and fast approximations to calculate the individual CLRs, thereby reducing the computational complexity.

In Chapter 5, we presented a real-time algorithm for estimating the Markov modulated Poission process parameters so as to calculate CLRs ranging over very low values, e.g., 1.0e-5 to 1.0e-8. The developed algorithm uses a fictitious queue and a window to determine the load state: underload or overload. Simulation of the performance of the algorithm showed that we can estimate the CLR closely by setting the window size to the minimum cell interval of the connections. We evaluated the effects of the connection peak rate, burstiness factor, and mean burst length on the accuracy of the estimated CLR. We found that our proposed method achieves conservative and accurate CLRs for all the values in which we are interested.

Because we measure the cell streams instead of measuring cell losses directly, we can estimate CLRs ranging over very low values accurately and quickly. Because cell streams are characterized as being in an overload state or underload state in our method, measurement algorithm is simple and thus suitable for online processing.

We concluded this dissertation by mentioning future work. The proposed measurement method in Chapter 3 should be extended into a network-wide bandwidth management framework. We are now designing such a framework [75, 76, 77, 78]. In the admission control for multiple QoS targets described in Chapter 4, we assume that accurate peak and average rates are given. However, it is difficult to declare accurate peak and average rates. A mechanism for measuring these rates and one for dynamic bandwidth re-allocation mechanism based on the measurements need to be developed.

Bibliography

[1] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 329–343, June 1993.

[2] N. Negroponte, *Being Digital*, Alfred A. Knopf, 1995.

[3] M. DePrycker, *Asynchronous Transfer Mode*, Ellis Horwood, Chichester, England, 1993.

[4] S. Keshav, *An Engineering Approach for Computer Networking*, Addison Wesley, 1997.

[5] K. Genda and N. Yamanaka, "A 160 Gb/s ATM switching system using an internal speed-up crossbar switch," in *Proc. of IEEE Globecom '94*, 1994, pp. 123–133.

[6] J. S. Turner and N. Yamanaka, "Architectual choice in large scale ATM switches," Tech. Rep. WUCS-97-21, Washington University, May 1997.

[7] J. S. Turner, "New directions in communications (or which way to the information age)," *IEEE Communications*, vol. 24, no. 10, pp. 8–15, Oct. 1986.

[8] ITU-T I. 371, "Traffic control and congestion control in B-ISDN," Geneva, Switzerland, Aug. 1996.

[9] ATM Forum, "Traffic management 4.0," Apr. 1996.

[10] ITU-T, "Recommendation Q. 2931," 1994.

[11] H. Saito and K. Shiomoto, "Dynamic call admission control in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 982–989, Sept. 1991.

[12] H. Saito, *Teletraffic technologies in ATM networks*, Artech house, Boston/London, 1994.

[13] "`http://www.mit.edu/people/mkgray/net`," .

[14] K. Shiomoto, M. Murata, Y. Oie, and H. Miyahara, "Performance evaluation of cell bypass queueing discipline for buffered Banyan type ATM switches," in *Proc. of IEEE INFOCOM '90*, pp. 677–685, San Francisco, 1990.

[15] K. Shiomoto, M. Murata, Y. Oie, and H. Miyahara, "Performance analysis of Banyan networks with pass ahead conflicting cells function for ATM switches," *IEICE Trans. Commun*, vol. J72-B-I, no. 11, pp. 1044–1054, Nov. 1989.

[16] K. Shiomoto, S. Chaki, and N. Yamanaka, "A simple bandwidth management strategy based on measurements of instantaneous virtual path utilization in ATM networks," to appear in IEEE/ACM Trans. Networking, Oct. 1995.

[17] K. Shiomoto and N. Yamanaka, "A simple multi-QoS ATM buffer management scheme based on adaptive admission control," in *Proc. of IEEE Globecom '96*, pp. 447–451, London, 1996.

[18] K. Shiomoto and N. Yamanaka, "An admission control scheme based on measurements of instantaneous utilization," To appear in IEICE Trans. Commun., Dec. 1997.

[19] K. Shiomoto, N. Yamanaka, and H. Hasegawa, "A low-pass filter design for ATM traffic measurement and its application to bandwidth management," preprint, Oct. 1997.

[20] K. Shiomoto and S. Chaki, "Adaptive connection admission control using real-time traffic measurements in ATM networks," *IEICE Trans. Commun.*, vol. E78-B, no. 4, pp. 458–464, Apr. 1995.

[21] K. Shiomoto and S. Chaki, "Performance evaluation of adaptive connection admission control using real-time traffic measurements in ATM networks," in *Proc. of IEEE BSS '95*, pp. 9–18, Poznań, Poland, 1995.

[22] C. Blondia, "The $N/G/1$ finite capacity queue," *Commun. Stat. -Stochast. Models*, vol. 5, no. 2, pp. 273–294, 1989.

[23] H. Yamada and F. Machihara, "Analysis and application of queueing model with phase type Markov renewal input," *IEICE Transaction*, vol. J73-B-I, no. 3, pp. 170–178, Mar. 1990.

[24] H. Yamada and S. Sumita, "A traffic measurement method and its application for cell loss probability estimation in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 315–324, Apr. 1991.

[25] A. Baiocchi and N. B. Melazzi, "Steady-state analysis of the MMPP$/G/1/K$ queue," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 531–534, Apr. 1993.

[26] Y. C. Jenq, "Performance analysis of a packet switch based on sigle-buffered Banyan network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, no. 12, pp. 1014–1021, Dec. 1983.

[27] H. S. Kim, "Performance of buffered Banyan networks under nonuniform traffic patterns," M.S. thesis, Department of Electrical Engineering, Universiry of Toronto, Dec. 1988.

[28] H. S. Kim and A. Leon-Garcia, "Performance of buffered Banyan networks under nonuniform traffic patterns," in *Proc. of IEEE INFOCOM'88*, pp. 4A.4.1–4A.4.10, 1988.

[29] M. G. Hluchyj and M. J. Karol, "Queueing in high-performance packet switching," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1587–1597, Dec. 1988.

[30] R. G. Bubenik and J. S.Turner, "Performance of a broadcast packet switch," *IEEE Trans. Commun.*, vol. 37, no. 1, pp. 60–69, Jan. 1989.

[31] G. Pujolle, "The influence of protocols on the stability conditions in packet-switching netowrks," *IEEE Trans. Commun.*, vol. 27, no. 3, pp. 611–619, Mar. 1979.

[32] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 833–846, Sept. 1986.

[33] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 856–868, Sept. 1986.

[34] B. Maglaris, D. Anastassiou, G. Karlsson, and J. Robbins, "Performance analysis of statistical multiplexing for packet video sources," in *Proc. of IEEE Globecom'87*, pp. 47.8.1–47.8.10, 1987.

[35] W. Verbeist, L. Pinno, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE J. Select. Areas Commun.*, vol. SAC-6, pp. 1623–1632, Dec. 1988.

[36] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Trans. on Networking*, vol. 4, no. 1, pp. 40–48, Feb. 1996.

[37] R. Gusella, "Characterizing the variability of arrival process with indexes of dispersion," *IEEE J. Select. Areas Commun.*, vol. 9, no. 2, pp. 203–211, Feb. 1991.

[38] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. of ACM SIGCOMM '93*, pp. 183–193, 1993.

[39] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, no. 3, pp. 226–244, June 1995.

[40] T. E. Tedijanto and L. Gun, "Effectiveness of dynamic bandwidth management mechanisms in ATM networks," in *Proc. of IEEE Infocom '93*, pp. 358–367, 1993.

[41] R. J. Gibbens, F. P. Kelly, and P. B. Key, "A decision-theoretic approach to call admission control in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1101–1114, Aug. 1995.

[42] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," in *Proc. of ACM SIGCOMM '95*, pp. 2–13, 1995.

[43] S. Shioda and H. Saito, "Real-time cell loss ratio estimation and its applications to ATM traffic controls," in *Proc. of IEEE Infocom '97*, pp. 1074–1081, 1997.

[44] Z. Dziong, M. Juda, and L. G. Mason, "A framework for bandwidth management in ATM networks – aggregate equivalent bandwidth estimation approach," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 134–147, Feb. 1997.

[45] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 56–70, Feb. 1997.

[46] S. Chaki, H. Saito, K. Miyake, and H. Ohnishi, "ATM network for high-speed data communication," in *Proc. of IEEE SICON/ICIE'93*, pp. 123–127, Singapore, 1993.

[47] N. Miyaho, M. Hirano, Y. Takagi, K. Shiomoto, and T. Takahashi, "An ATM switching system architecture for first generation of broadband services," in *Proc. of ISS '92*, pp. 285–289, Yokohama, 1992.

[48] J. Y. Hui, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, pp. 1598–1608, Dec. 1988.

[49] W. Fischer, E. Wallmeier, T. Worster, S. P. Davis, and A. Hayter, "Data communications using ATM: architectures, protocols, and resource management," *IEEE Commun. Mag.*, vol. 32, no. 8, pp. 24–33, Aug. 1994.

[50] J. W. Roberts, "Variable-bit-rate traffic control in B-ISDN," *IEEE Commun. Mag.*, vol. 29, no. 9, pp. 50–56, Sept. 1991.

[51] A. Baiocchi, N. B. Melazzi, M. Listanti, A. Roveri, and R. Winkler, "Loss performance analysis of an ATM multiplexer loaded with high-speed on–off sources," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 388–393, Apr. 1991.

[52] T. Murase, H. Suzuki, S. Sato, and T. Takeuchi, "A call admission control scheme for ATM networks using a simple quality estimate," *IEEE J. Select. Areas Commun.*, vol. 9, no. 9, pp. 1461–1470, Dec. 1991.

[53] S. Q. Li and C. L. Hwang, "Queue response to input correlation functions: discrete spectral analysis," *IEEE/ACM Trans. Networking*, vol. 1, no. 5, pp. 522–533, Oct. 1993.

[54] S. Q. Li and C. L. Hwang, "Queue response to input correlation functions: continuous spectral analysis," *IEEE/ACM Trans. Networking*, vol. 1, no. 6, pp. 678–692, Dec. 1993.

[55] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 325–334, Apr. 1991.

[56] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, Prentice Hall, 2 edition, 1997.

[57] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Baltimore, MD: John Hopkins Univ. Press, 1981.

[58] L. Kleinrock, *Queueing Systems*, vol. I, New York: John Wiley, 1975.

[59] R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, 1960.

[60] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 968–981, Sept. 1991.

[61] H. Esaki, K. Iwamura, T. Kodama, and T. Fukuda, "Connection admission control in ATM networks," *IEICE Trans. Commun.*, vol. E77-B, no. 1, pp. 15–27, Jan. 1994.

[62] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1566–1579, Feb./Mar./Apr. 1995.

[63] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," in *Proc. of ACM SIGCOMM '95*, pp. 100–113, 1995.

[64] H. Saito, "Call admission control in an ATM network using upper bound cell loss probability," *IEEE Trans. Commun.*, vol. 9, no. 40, pp. 1512–1521, Sept. 1992.

[65] G. Kesidis, J. Walrand, and C-S. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 424–428, Aug. 1993.

[66] E. Oki, N. Yamanaka, and K. Shiomoto, "A multi-QoS connection admission control scheme with ATM priority queueing system," in *Proc. of IEEE ICCCN '96*, pp. 275–276, Rockville, 1996.

[67] Y. Takagi, S. Hino, and T. Takahashi, "Priority assignment control of ATM line buffers with multiple QOS classes," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1078–1092, Sept. 1991.

[68] J. M. Hyman, A. A. Lazar, and G. Pacifici, "Real-time scheduling with quality of service constraints," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1052–1063, Sept. 1991.

[69] M. G. Hluchyj and A. Bhargava, "Queueing disciplines for integrated fast packet networks," in *Proc. of IEEE ICC '92*, pp. 335A.2.1–335A.2.7, 1992.

[70] K. Kang, Y. Yoon, and C. Kim, "A CAC scheme for heterogeneous traffic in ATM networks to support multiple QoS requirements," in *Proc. of IEEE Globecom '95*, pp. 422–426, 1995.

[71] H. J. Chao, "A novel architecture for queue management in the ATM network," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1110–1118, Sept. 1991.

[72] T. Kozaki, A. Takase, J. Yanagi, S. Tanabe, and S. Gohara, "ATM switching system with bandwidth control function," *IEICE Trans. Commun.*, vol. J76-B-I, no. 11, pp. 801–808, Nov. 1993.

[73] Y. Miyao, "A call admission control scheme in ATM networks," in *Proc. of IEEE ICC'91*, pp. 391–396, 1991.

[74] K. Noguchi and T. Okada, "Resource management strategy in an ATM network," *IEICE Trans. Commun.*, vol. J73-B-I, no. 10, pp. 733–743, Oct. 1990.

[75] N. Yamanaka, K. Shiomoto, and H. Hasegawa, "ALPEN: an ATM multiprotocol emulation network based on periodical performance check between edge node," in *Proc. of IEEE ICC '96*, pp. 392–397, Dallas, 1996.

[76] N. Yamanaka, K. Shiomoto, and H. Hasegawa, "ALPEN: An ATM multi-protocol emulation network, based on periodical performance check between edge nodes," *IEICE Trans. Commun.*, vol. J80-B-I, no. 7, pp. 542–553, July 1997.

[77] H. Hasegawa, N. Yamanaka, and K. Shiomoto, "Rate control for high-speed burst transmission with quick response in universal ATM-WAN, ALPEN," in *Proc. of IEEE ICC '97*, pp. 1734–1738, Montréal, 1997.

[78] H. Hasegawa, N. Yamanaka, and K. Shiomoto, "Architecture of ATM nodes with light-weight flow control for high-speed, multi-pprotocol ATM-WAN," in *Proc. of ISS '97*, pp. 443–449, Toronto, 1997.

Appendix A

Derivation of transition probabilities for analysis of Banyan network

A.1 Derivation of transition probabilities for 2×2 switching element

The derivation of transition probabilities for the 2×2 switching element is rather complicated to write down all of those here. Then, we will show an example for the derivation of the transition probability in this appendix. For this purpose, we consider the transition probability from the state $\mathbf{\Pi}(t) = ((1,2);1;\ (1,1);0)$ to the state $\mathbf{\Pi}(t+1) = ((1,2);1;\ (1,1);0)$ for the case of $S_1 = S_2 = 2$. The current state $\mathbf{\Pi}(t) = ((1,2);1;\ (1,1);0)$ is illustrated in Fig.A.1. The number specified in the buffer shows the destination output port for cells. The destination of the cell located in the third place in the input port buffer 1 has never been determined by our state representations. The cell bypass queueing discipline chooses the cell destined for output port 2 in the input port buffer 1, and the cell destined for the output port 1 at the head of the input port buffer 2 for the next transmissions. Depending on the condition of the output ports, the state transition can be classified into the following four cases.

Case 1: Only the output port 1 is available, i.e., next transmitting cell will be able to be accepted at the input port buffer of the next stage connected from output port 1 at the current stage. The corresponding probability is given by $r_1(1-r_2)$.

Only the cell at the input port buffer 2 can be transmitted to the output port, and the cell in the input port buffer 1 will be held at least until next time slot. Then the next state becomes $\mathbf{\Pi}(t+1) = ((1,2);1;(1,1);0)$ when (1) no cell arrives at input port 1 with probability $(1-(p_{11}+p_{12}))$, and (2) the cell destined for output port 1 arrives at input port 2 with probability p_{21}.

Case 2: Only the output port 2 is available. The probability of this event is given by $(1-r_1)r_2$.

Only the cell destined for output port 2 at input port buffer 1 can be transmitted. At input port buffer 1, the cell moving into the scope of the scanning range from the third location in the buffer should be destined for output port 2. Furthermore, a new cell should arrive for the next state $\mathbf{\Pi}(t+1)$

to be $((1,2);1;\ (1,1);0)$. Those probabilities are given by $p_{12}/(p_{11}+p_{12})$ and $p_{11}+p_{12}$, respectively. Furthermore, there should be no cell arrivals at input port buffer 2. This event occurs with probability $(1-p_{21}-p_{22})$.

Case 3: Both output ports 1 and 2 are available with probability $r_1 r_2$.

In this case, cells at both input ports 1 and 2 are served. Then, it is necessary that at input port 1, the cell moving into the scanning range is destined for output port 2, and that a cell newly arrives. A cell destined for output port 1 should arrive at input port buffer 2.

Case 4: Neither output port 1 nor output port 2 is available with probability $(1-r_1)(1-r_2)$.

In this case, no cells should arrive at both input port buffers 1 and 2. The probability for this event is given by $(1-(p_{11}+p_{12}))(1-(p_{21}+p_{22}))$,

By summing up all probabilities in the above four cases, we have

$$\begin{aligned} Prob[\mathbf{\Pi}(t+1)|\mathbf{\Pi}(t)] &= r_1(1-r_2)(1-p_{11}-p_{12})p_{21} \\ &\quad +(1-r_1)r_2p_{12}(1-p_{21}-p_{22}) \\ &\quad +r_1r_2p_{12}p_{21} \\ &\quad +(1-r_1)(1-r_2)(1-p_{11}-p_{12})(1-p_{21}-p_{22}). \end{aligned} \tag{A.1}$$

For other state transitions, the corresponding transition probability can be derived in a similar way.

A.2 Derivation of transition probabilities for IBC buffers

The transition probabilities for the l-th IBC buffer are given as follows:

For $q_{\mathrm{IBC}_l}(t)=0$

$$q_{\mathrm{IBC}_l}(t+1)=\begin{cases} 0 & \text{with prob. } 1-p_{\mathrm{IBC}_l} \\ 1 & \text{with prob. } p_{\mathrm{IBC}_l} \end{cases} \tag{A.1}$$

For $1 \le q_{\mathrm{IBC}_l}(t) < L_{\mathrm{IBC}_l}$

$$q_{\mathrm{IBC}_l}(t+1)=\begin{cases} l-1 & \text{with prob. } (1-p_{\mathrm{IBC}_l})r_{\mathrm{IBC}_l} \\ l & \text{with prob. } (1-p_{\mathrm{IBC}_l})(1-r_{\mathrm{IBC}_l})+p_{\mathrm{IBC}_l}r_{\mathrm{IBC}_l} \\ l+1 & \text{with prob. } p_{\mathrm{IBC}_l}(1-r_{\mathrm{IBC}_l}) \end{cases} \tag{A.2}$$

For $q_{\mathrm{IBC}_l}(t)=L_{\mathrm{IBC}_l}$

$$q_{\mathrm{IBC}_l}(t+1)=\begin{cases} L_{\mathrm{IBC}_l}-1 & \text{with prob.}(1-p_{\mathrm{IBC}_l})r_{\mathrm{IBC}_l} \\ L_{\mathrm{IBC}_l} & \text{with prob.}(1-p_{\mathrm{IBC}_l})(1-r_{\mathrm{IBC}_l})+p_{\mathrm{IBC}_l}r_{\mathrm{IBC}_l}+p_{\mathrm{IBC}_l}(1-r_{\mathrm{IBC}_l}) \end{cases} \tag{A.3}$$

Appendix B

Power spectral density function for low-pass filter

B.1 Sliding window

A z-transform of Eq. (3.4) becomes

$$\lambda(z)=\frac{1}{w}\sum_{k=0}^{T-1} z^{-k}a(z). \tag{B.1}$$

Now the transfer function of Eq. (B.1) $H(z)$ becomes

$$H(z)=\frac{\lambda(z)}{a(z)}=\frac{1}{T}\sum_{k=0}^{T-1} z^{-k}. \tag{B.2}$$

Substitute z by $\exp(j\omega T_s)$ in Eq. (B.2), we have

$$\begin{aligned} H(e^{j\omega T_s}) &= \frac{1}{w}\sum_{k=0}^{T-1} e^{-jk\omega T_s} \\ &= \frac{1}{T}\sum_{k=0}^{T-1}(\cos(k\omega T_s)-j\sin(k\omega T_s)), \end{aligned} \tag{B.3}$$

where $j=\sqrt{-1}$ and $\omega=2\pi f$,f denotes the signal frequency. Eq. (B.4) can be expressed by the amplitude characteristics $|H(\omega)|$ and the phase characteristics $\theta(\omega)$.

$$H(e^{j\omega T_s})=|H(\omega)|e^{j\theta(\omega)}. \tag{B.4}$$

Now a power spectral function of Eq. (3.4) $S(\omega)$ is given by Eq. (B.5).

$$S(\omega)=\frac{1}{T^2}\sum_{k=0}^{T-1}(\cos^2(k\omega T_s)+\sin^2(k\omega T_s)). \tag{B.5}$$

B.2 Recursive low-pass filter

A z-transform of Eq. (3.6) becomes

$$\lambda(z) = \alpha a(z) + (1-\alpha)z^{-1}\lambda(z). \tag{B.6}$$

The transfer function of Eq. (B.6) $H(z)$ is given by

$$H(z) = \frac{\lambda(z)}{a(z)} = \frac{\alpha}{1-(1-\alpha)z^{-1}}. \tag{B.7}$$

Substitute z by $\exp(j\omega T_s)$ in Eq. (B.7),

$$H(e^{j\omega T_s}) = \frac{\alpha}{1-(1-\alpha)\exp(-j\omega T_s)}. \tag{B.8}$$

Eq. (B.8) is expressed by the amplitude characteristic $|H(\omega)|$ and the phase characteristics $\theta(\omega)$.

$$H(e^{j\omega T_s}) = |H(\omega)|e^{j\theta(\omega)}. \tag{B.9}$$

Power spectral function of Eq. (3.6) $S(\omega)$ is given by Eq. (B.10).

$$S(\omega) = |H(\omega)|^2 = \frac{\alpha^2}{1+(1-\alpha)^2-2(1-\alpha)\cos(\omega T_s)}. \tag{B.10}$$