Studies on Effective Data Transfer Mechanisms for Future High-Speed Networks

Shingo Ata

February 2000

Department of Informatics and Mathematical Science
Graduate School of Engineering Science
Osaka University

Preface

A rapid spread of the Internet requires ISPs (Internet Service Providers) to increase their bandwidth. However, bandwidth increasement only is insufficient to provide stable QoS (Quality of Service)–rich communication services; we have to consider additional issues to effectively utilize the available bandwidth. One important example is traffic management such as congestion control, bandwidth reservation, and data retransmission. In best–effort networks such as the conventional Internet, all connections passing through the link are affected by even an instant occurrence of congestion. As a result, the utilization of network bandwidth is degraded because of packet losses, and not a little bandwidth is wasted. With the adequate traffic management, it is expected to catch up such a wasted bandwidth. Another factor is to speed up the routers to follow up the growth of link bandwidth. Due to hardware specifications and current packet forwarding mechanisms of routers, it is considered that the performance of routers is limited. It shows the necessity of considering some alternative techniques to solve the performance limitation.

Keeping those facts in mind, we investigate an effective data transfer mechanisms in high–speed networks. We mainly consider two issues; one is the traffic management in high–speed networks. The other is the new architecture of high–speed routers.

For the first issue, we focus on high–speed ATM networks where several service classes is defined in the standards to guarantee various types of end–to–end QoS. Among service classes, ABT (ATM Block Transfer) service class is promising for an effective data transfer because the bandwidth reservation is performed on

burst–basis and once the reservation is permitted, the cell loss does not occur within the network. There are two variants in ABT; ABT/IT (Immediate Transmission) and DT (Delayed Transmission). We first develop an approximate analysis to discuss performance comparisons of ABT/IT and DT. Through numerical examples, we show that ABT/DT is quite sensitive to propagation delays while ABT/IT is not. We next propose performance improvement of ABT/DT with dynamic bandwidth negotiation. Simulation results show that the bandwidth negotiation can improve the throughput of ABT/DT and it outperforms even ABT/IT when propagation delay is relatively small at the expense of the increased burst transmission times. We further examine the effect of bandwidth reduction methods at the backoff in both of ABT/DT and IT, and show that a flexible use of the bandwidth can be achieved in both protocols, leading to fairly good performance. However, we also find that the throughput of ABT/IT is drastically decreased when the traffic load becomes heavy and/or the number of hops of the connection becomes large.

To avoid this performance degradation in ABT/IT, we next propose a new protocol, *Buffered ABT/IT*, which makes reservation on the buffer as well as the bandwidth. The approximate analysis method is developed for buffered ABT/IT. Through numerical examples, we show that ABT/IT with buffer reservation can lead to much performance improvement comparing with the original one. We also address the required buffer capacity for obtaining high throughput, and show that high performance can be achived by a reasonable buffer size even in the current memory technology. We next consider the bandwidth negotiation mechanism described above and show the performance improvement from the original ABT/IT protocol. However, in the case of the buffered–ABT/IT, it already provides high throughput and the performance improvement by the bandwidth negotiation mechanism is quite small.

Then we consider the upper layer transport protocol since it also provide a congestion control mechanism and such redundant controls may incur performance degradation unexpectedly. Through performance evaluation on TCP (Transmission

Control Protocol) over ABT protocol stack, we show that the retransmission mechanism of ABT can effectively overlay the TCP congestion control mechanism so that TCP operates in a stable fashion and works well only as an error receovery mechanism.

For the second issue, we investigate appropriate parameter tuning method in high–speed router by considering traffic characterstics. We first analyze the actual network traffic gathered by the traffic monitor. Through the statistical analysis, we show that the number of packets in the flow and active flow durations follow the log–normal distributions, and the tails follow the pareto distribution, which is known as a class of the heavy–tailed distribution. We next show the determination method for control parameters in high–speed MPLS routers based on those observations. Through simulation experiments, it is shown that VCs are stably and highly utilized by applying the analytical results on traffic characteristics. We also show the effect of flow aggregation, in which flows are aggregated into one with a larger granularity of classification. The simulation results show that our flow aggregation can give a significant impact on the performance of MPLS routers.

Acknowledgments

I would like to express my sincere appreciation to Prof. Masayuki Murata of Osaka University, my adviser, for his innumerable help and continuous support. I am heartily grateful to Prof. Toshinobu Kashiwabara, and Prof. Akihiro Hashimoto for serving as readers of my dissertation committee. Their expertise and insightful comments have been invaluable.

I am most grateful to Prof. Hideo Miyahara for his encouragement and invaluable comments in preparing this dissertation. This dissertation would not have been possible without his guidance and inspiration.

All works of this dissertation would not have been possible without the support of Associate Prof. Tetsuya Takine of Kyoto University. It gives me great pleasure to acknowledge his assistance. He has been constant sources of encouragement and advice through my studies and preparation of this manuscript.

I am also indebted to Associate Prof. Ken-ichi Baba, Assistant Prof. Naoki Wakamiya, Dr. Hiroyuki Ohsaki of Osaka University who gave me helpful comments and feedbacks.

I thank many friends and colleagues in the Department of Informatics and Mathematical Science of Osaka University for their support — special thanks to Mr. Go Hasegawa and Mr. Kentarou Fukuda for their expert suggestions as well as warmheartedness.

I am deeply grateful to my parents. They always give me endless love. Finally, my heartfelt thanks to my wife Junko for her endless patience, encouragement, and understanding with my all love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

A rapid growth of the Internet and proliferation of new multimedia applications lead to demands of high speed and broadband network technologies. Accordingly, the network capacity has been increased rapidly. In Japan, for example, several ISPs (Internet Service Provider) increase the backbone bandwidth from 1.5 Mbps to 155 Mbps in last two years. However, to get rid of end user's complaints, the solution to speed up the link is not sufficient because of following reasons; First, the required bandwidth depends not only on the number of users (i.e., hosts connected to the Internet) but also on the kind of applications. For example, the required bandwidth differs among an E–Mail and a motion video or another multimedia applications. When some people begin to watch videos via the Internet, the network can easy be congested. To solve this problem, the traffic management is effective. In best–effort networks, all connections passing through the link are affected by even an instant of congestion. As a result, the utilization of network bandwidth is degraded because of packet losses, and not a little bandwidth is wasted. With the traffic management, it is expected to catch up such a wasted bandwidth. When the network is congested, it has a great impact to high–speed networks. Second, because

of a drastic progress in the link capacity, another kind of network resources such as routers can be a bottleneck of the network performance. Due to hardware specifications and current packet forwarding mechanisms of routers, it is considered that the performance of routers is limited. It shows the necessity of considering some alternative techniques to solve the performance limitation.

Keeping those facts in mind, we investigate an effective data transfer mechanisms in high–speed networks. For this purpose, we mainly consider two issues; one is the traffic management in high–speed networks. For an underlying network, we focus on ATM (Asynchronous Transfer Mode) which support not only high–speed link capacity with an optical technology, but also service classes to guarantee various end–to–end QoS such as real time traffic or data transmissions. The other is the architecture of high–speed routers to follow up the growth of link bandwidth.

1.2 ABT Service Class in ATM Networks

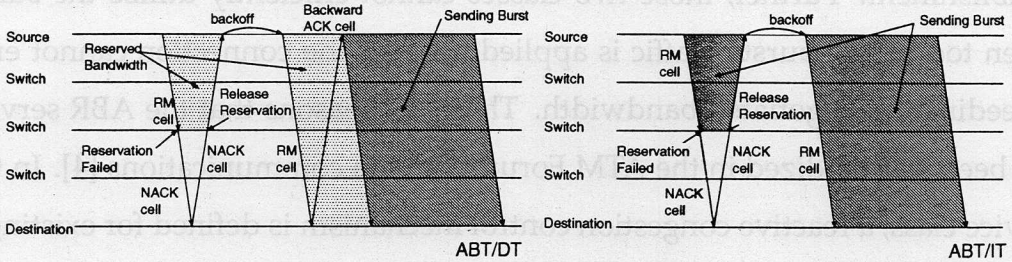
ATM has been developed as a network technology to integrate various media, including a real–time traffic like video and audio as well as non–real–time traffic such as text and image, through a unified interface. A traffic management is a key technique to guarantee the user's required QoS (Quality of Service) and keep an efficient use of network resources. In two standardization bodies, ITU–T and ATM Forum, several service classes have been defined according to the QoS of the multimedia traffic [1-3]. Those are CBR (Constant Bit Rate), VBR (Variable Bit Rate), ABR (Available Bit Rate) and UBR (Unspecified Bit Rate) service classes. Note that in the new ITU–T document [1], CBR and VBR are now called as DBR (Deterministic Bit Rate) and SBR (Statistical Bit Rate), respectively.

In CBR and VBR service classes, a fixed amount of resources is actually or virtually allocated to each connection to support real–time communications such as motion video and audio transmission [3]. To guarantee QoS in those service classes, however, the source must know traffic characteristics in advance of the connection

establishment. Further, those two classes cannot efficiently utilize the bandwidth when too highly bursty traffic is applied because the connection cannot emit cells exceeding the negotiated bandwidth. Those are reasons that the ABR service class has been standardized in the ATM Forum for data communications [4]. In the ABR service class, a reactive congestion control mechanism is defined for existing bursty data communications, and its main concern is to guarantee a bound on cell loss ratio. However, careful parameter tuning is necessary to guarantee no cell loss, and a set of optimal control parameters can be chosen only when the number of active connections are fixed or accurately be estimated [4]. The UBR service class does not guarantee any QoS parameter.

Another service class, ABT (ATM Block Transfer) [1, 5, 6] service class which is defined by ITU–T, is also intended to be applied to data communications. A difference from the ABR service class is that the bandwidth is explicitly reserved before cell emission. In that sense, the ABT service class is similar to CBR/VBR service classes. However, bandwidth reservation is not performed at the connection setup time, but deferred to time when the burst actually arrives at the source. Here, the burst means the data unit of ABT. For example, the burst may correspond to the file or block in the case of file transfer. Once the bandwidth reservation is admitted by the network, the source can transmit the burst with no cell loss. Network resources can be highly utilized without knowledge of traffic characteristics and ABT is expected to be suitable to bursty data transfer. Note that in ITU–T, ABT was called as FRP (Fast Reservation Protocol) before the standardization and is still under study [1]. On the other hand, in the ATM Forum, ABT is not yet fully addressed since much efforts have been focused on the definition of ABR service class in the traffic management working group.

In the original ABT protocol [1, 5-9], only routing is performed at the connection setup time, and the bandwidth is not reserved. When the burst arrives at the source, a forward RM (Resource Management) cell is sent to the destination along the predetermined route for the bandwidth reservation. The RM cell contains the required



(a) ABT/DT (b) ABT/IT

Figure 1.1: ABT Protocols

bandwidth to transfer the burst. Each switch on the route reserves the bandwidth and then forwards the RM (ACK) cell to the next switch in the downstream. If a sufficient bandwidth is not available on the link, on the other hand, the switch directly forwards the RM (NACK) cell to the destination. The destination then returns the RM (ACK) or RM (NACK) cell to the source as the backward RM cell. Every switch receiving the backward RM (NACK) cell releases the reserved bandwidth if it has been reserved. The source receiving the backward RM cell can finally recognize whether the bandwidth request is admitted or not. The protocol stated above is called ABT with Delayed Transmission (ABT/DT), which is illustrated in Fig. 1.1(a).

When the link bandwidth becomes large as in recent high speed networks, the overhead time to wait the backward RM cell before the burst transmission is not acceptable. That is the reason why ABT/IT (ABT with Immediate Transmission) is introduced [5]. In ABT/IT, the source sends the burst immediately following the forward RM cell without waiting for acknowledge of the reservation as shown in Fig. 1.1(b). If the sufficient bandwidth is available, each switch accepts the burst and forwards it to the next switch. If not, on the other hand, the switch discards the incoming burst, and returns the backward RM (NACK) cell via the destination to notify the source that the burst is lost. While this mechanism introduces a hardware complexity to selectively discard the burst, it must alleviate the influence of the large

propagation delay.

1.3 Effective Resource Reservation Mechanisms in ABT Protocols

In the above, we have assumed that each switch reserves the bandwidth as specified in the forward RM cell. In the case of ABT/DT, however, it is possible that the switch reserves the bandwidth less than the requested bandwidth according to the link condition. The switch then replaces the requested bandwidth on the RM cell with the reserved and forwards it to the next switch. In this bandwidth reservation method, each switch receiving the backward RM cell confirms the bandwidth reservation with the bandwidth specified in the RM cell. When the source receives the backward RM (ACK) cell, it begins burst emission according to the specified bandwidth. Note that only ABT/DT can employ this mechanism because the source emits the burst before receiving the acknowledgement.

An alternative way to allow in ABT/IT flexible bandwidth reservation may be implemented in the backoff algorithm. When the source fails the bandwidth reservation, it re–tries the reservation after some time interval, which is called "backoff". The bandwidth to reserve is reduced in the retrials expecting that the reduced bandwidth becomes smaller than the available bandwidth on the most congested link and will be accepted.

With those mechanisms mentioned above, the effective resource utilization can be accomplished. However, there are two kinds of resources within the network, the bandwidth and the buffer. The ABT protocol only reserves the bandwidth for transferring the burst, because the buffer size of the ATM switch was limited when the original ABT protocol was proposed [5, 10]. The output port buffer of a hundred of cells was common. However, a recent advancement of the switching technology allows the ATM switch to be equipped with a buffer of thousand cells for each port.

Thus, we can now expect further performance improvement with ABT by reserving the buffer in addition to the bandwidth.

Then we consider the upper layer transport protocol since it also provide a congestion control mechanism and it may be claimed that such redundant controls incurs performance degradation.

TCP has two main functions; window–based flow control and error recovery control between two end systems and congestion control for the network. In particular, window–based flow control regulates end–to–end flow and retransmits the segments if segments are lost in the network. At the same time, the source indirectly monitors the network congestion status via round trip times of acknowledgments, and adjusts its window size.

Thus, if we employ ABT protocols below TCP, we have two independent congestion control mechanisms at distinct layers; the Transport Layer (TCP) and the ATM Layer (ABT). In such a situation, congestion control mechanisms may interact with each other in an ill fashion, and the performance will be unexpectedly degraded. Most important is thus to identify interactions of two congestion control mechanisms provided by two layers. An ideal solution to accomplish high performance data transfer would be that the ATM layer protocol manages congestion control while the transport layer protocol is devoted to the flow control and error recovery control between two end systems.

Through performance evaluation on TCP (Transmission Control Protocol) [11] over ABT protocol stack,we show that the retransmission mechanism of ABT can effectively overlay the TCP congestion control mechanism so that TCP operates in a stable fashion and works well only as an error recovery mechanism.

1.4 Data Transfer Mechanisms in High–Speed Routers

The increase of packet processing capability of routers is also indispensable to efficiently utilize the growing link bandwidth and several techniques have been pro-

posed for high speed routers. For example, IETF is now standardizing MPLS (Multi Protocol Label Switching) [12], which combines the flexibility of layer–3 routing with the high capacity of layer–2 switching.

In MPLS, the router identifies a *flow* by IP addresses and applications (i.e., port numbers) of packets, and forwards them through faster switching paths. To identify the flow to pass through switching paths, MPLS assigns an unique label (i.e., VCI/VPI in MPLS over ATM networks) for each flows. However, due to the hardware specification of the router, the number of labels (VCs) are limited. Moreover, the performance of MPLS routers is measured by how many packets can be transmitted through hardware switching (i.e., VCs). From these reasons, MPLS switches are necessary to identify flows containing a large number of packets, which we will refer to as long–lived flows.

In MPLS, two parameters are used to identify long–lived flows. The first parameter is the threshold value X; the MPLS switch monitors the number of packets for each flow, and determines the flow with X or more packets as a long–lived flow. The other parameter is a time–out value T. Since the MPLS switch cannot detect the end of flow, the switch decides that the flow is finished when the packet does not arrive on the assigned VC during T seconds. If the parameter X is small, many VC assignments would be required not only for long–lived flows (which includes the large number of packets in the flow) but also for short–term flows, which results in the failure of setting more VCs for the long–lived flows newly arriving at the switch. On the other hand, if the parameter X is large, the utilization of VC space becomes low, and the switch performance is degraded. Moreover, the larger value of the parameter T leads to lower utilization of VC spaces while the switch with the small parameter T releases a VC of an active flow.

1.5 Related Works

The authors in [13] compared ABT/IT and DT with an arbitrary network topology, and observed that ABT/DT is sensitive to propagation delays. However, the comparison is not fair because the model in [13] assumed that the switch which received the backward RM (NACK) cell still holds its reserved bandwidth in ABT/IT.

The burst scheduling networks, which is proposed in [14, 15], is a new class of packet switching networks and seems preferable to the bursty traffic. Its objective is to guarantee the end-to-end delay and delay jitter for efficiently transferring the bursty data. More recently, they notice that their architecture is similar to ABT, and in [16], they described their architecture in the context of ABT protocols. In their paper, however, they only focused on the implementation issue of the admission control algorithm based on ABT to guarantee the burst transfer delay and loss rate, and they did not take into account a possibility of failures of the bandwidth reservation requests. On the other hand, we develop an approximate analysis of blocking probability of bursts in ABT/IT and DT.

Other studies on ABT can be found in [17-19], but those are limited to ABT/DT. In [17, 18], the authors showed the effect of the dynamic bandwidth reduction in ABT/DT on the single link model. In [19], they employed the network model with an arbitrary topology to verify that observations in [17, 18] can also be applied to general networks. However, they assumed the propagation delay was negligible [19]. On the contrary, in this dissertation, we first investigate the basic performance of ABT/IT and DT for the arbitrary network with non–zero propagation delay by developing the approximate analytic approach. We then show how bandwidth is effectively and flexibility utilized in the bandwidth negotiation method in ABT/DT and backoff methods in both of ABT/DT and IT through simulation experiments.

Fast reservation protocols on the bandwidth and on the buffer was investigated separately. The ABT protocols mentioned above are examples of the fast bandwidth

reservation. The idea of the fast buffer reservation can be found in [20]. Another fast buffer reservation method was proposed in [21], which is known as a credit–based congestion control method. While it is shown to be effective for the local area network (LAN) environments with the small propagation delay, there is a limit on the applicability to the WAN environment. It is the reason that the credit–based congestion control was not adopted as a standard for the ABR service class by the ATM Forum [4]. A proposed approach in this dissertation is to incorporate the buffer reservation mechanism in the context of the ABT protocol, by which the performance improvement of the ABT protocol is expected.

Traffic monitoring on the actual Internet traffic and its statistical analysis was studied in literatures [22-26]. However, few applications were studied to which analytic results are applied. It is necessary in high speed switching routers to determine of control parameters from traffic characteristics.

Authors in [27] showed appropriate parameters in MPLS by using trace–driven simulator and Internet traffic archives. Such approach, however, requires much times of simulation for each traced data. On the contrary, our approach is based on statistical analysis and can be generalized for arbitrary traffic patterns. Effect of a parameter set (X, T) in MPLS was also studied [28], but this study did not take account of traffic characteristics.

1.6 Outline of Dissertation

Performance Comparisons of ABT/IT and DT [29-32]

First, in Chapter 2, the basic performance of ABT/IT and DT for the arbitrary network with non–zero propagation delay is investigated. It is then shown that the flexible bandwidth usage by virtue of the bandwidth negotiation method in ABT/DT and backoff methods in both of ABT/DT and IT can allow the performance improvements. Numerical results shows that ABT/IT is robust in the sense that its per-

formance is not much affected by the propagation delay compared with ABT/DT. When the traffic load becomes heavy, however, the throughput of even ABT/IT is drastically decreased. It is especially true for long–hop connections.

Performance Improvement of ABT Protocols with Combined Bandwidth/Buffer Reservation [33-36]

The buffered ABT/IT, which reserves the buffer as well as the bandwidth for burst transmission, is proposed in Chapter 3 to avoid the performance degradation of long–hop connections. We develop the approximate analysis to evaluate the throughput and the mean transfer delay of the buffered ABT/IT. Then, especially long–hop connections by comparing with the original ABT/IT protocol, we show that buffer reservation in ABT/IT can lead to much performance improvement. We also address the required buffer capacity for obtaining high throughput, and show that it can be realized with the current technology. Finally, we apply the bandwidth negotiation mechanism which improves the performance of the original ABT/IT protocol to buffered ABT/IT and show that the performance improvement by the bandwidth reduction mechanism is small.

Performance Evaluation of TCP over ABT Protocols [37-39]

In Chapter 4, we provide comparative evaluation of TCP over ABT protocols. In our experiments, we consider (1) ABT/IT, (2) buffered ABT/IT [35, 36], (3) ABT/DT, and (4) ABT/DT with bandwidth negotiation [31, 32] as underlying ATM layer protocols. As an alternative to ABT protocols, we also evaluate the performance of TCP over UBR with an EPD enhancement for comparison purpose. Our simulation results show that the retransmission mechanism of ABT can overlay the TCP congestion control mechanism so that TCP operates in a very stable fashion, and that TCP over ABT (especially TCP over buffered ABT/IT and TCP over ABT/DT with bandwidth negotiation) can outperform TCP over EPD in various network condi-

tions. However, we should note here that we had expected for TCP and original ABT protocols without any modifications to establish high performance data transfer, but it was not true. Actually, if we employ the original ABT protocols, achievable throughput becomes less than that in TCP over EPD, and it leads to a serious unfairness among connections of different hop counts.

Analysis of Network Traffic and its Application to Design of High–Speed Routers [40-44]

In Chapter 5, we first investigate the characteristics of the actual Internet traffic using the traffic monitor OC3MON developed by MCI [22]. When ATM is considered as a underlying network in MPLS, the MPLS router assigns the VC (virtual circuit) of ATM to each flow.

The number of packets in the flow depends on the parameter set (X, T) and it is impossible to consider those two parameters independently. We demonstrate how our statistical analysis can be utilized for determining those two parameters in order to obtain high performance MPLS routers. Since the traffic load (the arrival rate of flows) at the routers is time–varying in our traced data, we also show the adaptive control method to determine the parameters according to the traffic load.

One technique to effective utilize VC space is a flow aggregation, in which flows are aggregated into one with a larger granularity of classification (e.g., from port number to IP address). Aggregated flows have a larger number of packets and a longer flow duration. These properties give a significant impact on the performance of MPLS routers [26]. In this dissertation, we show that the aggregated flow has a same statistical distribution, which means that our analytic formulas can be applied to aggregated flows. Then, the effect of flow aggregation is demonstrated using the simulation technique.

Chapter 2

Performance Comparisons of ABT/IT and DT

In this chapter, We develop an approximate analysis for two types of ABT (ABT/DT and ABT/IT) to discuss performance comparisons of ABT/IT and DT. Through numerical examples, we show that ABT/DT is quite sensitive to propagation delays while ABT/IT is not. We next investigate performance improvement by dynamic bandwidth negotiation, which is applicable to ABT/DT. Bandwidth reduction methods in both of ABT/DT and IT are also examined. Simulation results show that the bandwidth negotiation can improve the performance of ABT/DT, and that in the case of short propagation delays, it outperforms even ABT/IT in terms of throughput. However, it is obtained at the expense of the increased burst transmission times. On the other hand, the bandwidth reduction allows a flexible use of the bandwidth, leading to fairly good performance in all parameter regions.

2.1 Analysis of ABT

2.1.1 Mathematical Model and Assumptions

Consider a network with J (> 0) links, labeled 1 to J. Link j ($j = 1, \ldots, J$) has capacity B_j (Mbps). Adjacent two links are connected by a node (i.e., an ATM switch). Every terminal is connected to the end node. We assume that every end node pair has a predefined route (i.e., fixed routing strategy), and that the ATM SVC (Switched Virtual Connection) has already been established along the route between every two terminals. Each end node has a number of terminals such that the arrivals of bursts can be assumed to follow a Poisson distribution. Then, we regard multiple SVCs on the route between two end node pair as an aggregated single *connection*. The number of asymmetric one-way *connections* in the network is denoted by P, and the bandwidth is not reserved for each connection until the burst is actually generated at the source of the connection according to the ABT protocol.

To analyze the performance of the above network, we assume the followings. All links is assumed to have the same length and we denote the round-trip propagation delay of a link by D. Let $J^{(p)}$ ($p = 1, \ldots, P$) denote a set of links on the route of connection p. For $j \in J^{(p)}$, let $J_{j+}^{(p)}$ (resp. $J_{j-}^{(p)}$) denote a set of links between the source (resp. the destination) and link j on the route. Thus, $J^{(p)} = J_{j+}^{(p)} \cup j \cup J_{j-}^{(p)}$ for $j \in J^{(p)}$. Let H denote the maximum number of links among all routes, i.e., $H = \max_p |J^{(p)}|$. Bursts on the pth connection are generated according to a Poisson process with rate λ_p. Lengths of bursts of all connections are independent and identically distributed according to an exponential distribution with mean $1/\mu$ (Mbit). By assuming that all bursts are transmitted with B Mbps, transmission times of all bursts are exponentially distributed with mean $(\mu b)^{-1}$ (sec). Let $R_j^{(h)}$ ($j = 1, \ldots, J, h = 1, \ldots, H$) denote a set of connections which has link j as the hth link from their destinations. Further, let R_j ($j = 1, \ldots, J$) denote a set of connections having link j on their routes, i.e., $R_j = \cup_{h=1}^{H} R_j^{(h)}$. We then assume $\sum_{p \in R_j} \lambda_p / \mu < B_j$ for all $j = 1, \ldots, J$, which ensures

that the network resources are ample for supporting all connections. We assume that the overhead of RM cell transmissions are negligible. We also assume that the network is stable and has reached its steady state in the rest of the section.

2.1.2 Analytical Approach

The mathematical model is considered as a variant of loss networks. However, since we explicitly model propagation delays, the network does not have the product-form solution. Thus we do not expect any solution methods to evaluate the throughput performance exactly. We therefore provide an approximate analysis to obtain the throughput performance. We first adopt the reduced load approximation which is a common technique to analyze large-scale loss networks. The essential point in the reduced load approximation is to treat all links independently, while the influence of other links on the target link is taken into account by reducing the load on the target link. As for the reduced load approximation, readers are referred to [45-50].

By virtue of the reduced load approximation, the throughput θ_p (Mbps) of connection p is given by

$$\theta_p = \frac{\lambda_p}{\mu} \prod_{j \in J(p)} (1 - E_j), \tag{2.1}$$

where E_j denotes the blocking probability on link j. In what follows, we analyze the blocking probabilities in ABT/IT and ABT/DT separately.

2.1.3 Blocking Probabilities in ABT/IT

Let $\boldsymbol{n}_j$ denote the state of link j:

$$\boldsymbol{n}_j = (n_{j,1}, n_{j,2}, \ldots, n_{j,H}),$$

where $n_{j,1}$ denotes the number of bursts being successfully transmitted on link j and $n_{j,h}$ $(h = 2, \ldots, H)$ denotes the number of bursts in transmission on link j, whose connection is in $R_j^{(h)}$, while being failed in transmission in one of the downstream

links on the route. We define $h(\boldsymbol{n}_j)$ as the remaining amount of bandwidth given $\boldsymbol{n}_j$:

$$h(\boldsymbol{n}_j) = B_j - b\sum_{h=1}^{H} n_{j,h}.$$

Let S_j be

$$S_j = \{\boldsymbol{n}_j \mid h(\boldsymbol{n}_j) \leq B_j, n_{j,h} \geq 0\ (h = 1, \ldots, H)\}, \qquad j = 1, \ldots, J.$$

Note that S_j denotes a feasible set of states of link j.

We now consider holding times of bursts from connection $p \in R_j^{(h)}$ $(h = 2, \ldots, H)$ on link j, which fail in transmission in one of downstream links on the route. Note that the failure of the transmission is notified after time hD. Thus, if the transmission time of a failed burst is longer than hD, the holding time is given by hD. Otherwise, the holding time is identical to the transmission time. Therefore the distribution of holding times of failed bursts is given by a truncated exponential distribution with a mass at hD, whose mean μ_h^{-1} is given by

$$\mu_h^{-1} = \frac{1 - e^{-\mu bhD}}{\mu b}.$$

We define a $1 \times H$ unit vector $\boldsymbol{e}_h$ $(h = 1, \ldots, H)$ as

$$\boldsymbol{e}_h = (0, \ldots, 0, \underbrace{1}_{h\text{th}}, 0, \ldots, 0).$$

The transition from state $\boldsymbol{n}_j$ to state $\boldsymbol{n}_j + \boldsymbol{e}_1$ happens with rate $r_{j,1}$, where

$$r_{j,1} = \begin{cases} \displaystyle\sum_{p \in R_j} \lambda_p \prod_{k \in J_{j+}^{(p)}} (1 - E_k) \prod_{k \in J_{j-}^{(p)}} (1 - E_k), & \text{if } \boldsymbol{n}_j \in S_j, h(\boldsymbol{n}_j) > b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Note that empty products are defined to be one in the above and hereafter. Further the transition from state $\boldsymbol{n}_j$ to state $\boldsymbol{n}_j + \boldsymbol{e}_h$ $(h = 2, \ldots, H)$ happens with rate $r_{j,h}$, where, for $h = 2, \ldots, H$,

$$r_{j,h} = \begin{cases} \displaystyle\sum_{p \in R_j^{(h)}} \lambda_p \prod_{k \in J_{j+}^{(p)}} (1 - E_k) \left[1 - \prod_{k \in J_{j-}^{(p)}} (1 - E_k)\right], & \text{if } \boldsymbol{n}_j \in S_j, h(\boldsymbol{n}_j) > b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Note here that all bursts are generated according to Poisson processes. Therefore we approximate the arrival process from each connection to link j by a Poisson process and we aggregate all arrivals into one Poisson stream. The traffic intensity ρ_j of the aggregated stream in link j is given by

$$\rho_j = r_{j,1}(\mu b)^{-1} + \sum_{h=2}^{H} r_{j,h}\mu_h^{-1}.$$

Let N_j be

$$N_j = n_{j,1} + \ldots + n_{j,h}$$

It then follows from the insensitivity property of $M/G/c/c$ that [51]

$$\Pr(N_j = k) = \frac{\rho_j^k/k!}{\sum_{i=0}^{K_j} \rho_j^i/i!}, \qquad k = 0, 1, \ldots, K_j,$$

where K_j denotes the maximum integer which is not greater than B_j/b. Thus the blocking probability E_j on link j is given by

$$E_j = \Pr(N_j = K_j) = \frac{\rho_j^{K_j}/K_j!}{\sum_{i=0}^{K_j} \rho_j^i/i!}, \tag{2.4}$$

We now provide an iterative procedure to obtain the blocking probability E_j. In what follows, for any symbol X, we denote the value of X in the nth iteration by ${}_{(n)}X$.

Step 1. Initial input: for all $p = 1, \ldots, P$ and all $j = 1, \ldots, J$,

i) Let ${}_{(0)}E_j = 0$

ii) Let ${}_{(0)}\theta_p = \lambda_p/\mu$.

iii) Set a nonnegative small value to ϵ (e.g., $\epsilon = 10^{-3}$ for graphical representations).

iv) Let $n = 1$.

Step 2. The nth iteration:

i) Compute the arrival rates $r_{j,h}$ $(h = 1\ldots., H)$ in (2.2) a nd (2.3) for all $j = 1, \ldots, J$ with ${}_{(n-1)}E_j$.

ii) Compute the right hand side of (2.4) and let ${}_{(n)}E_j$ be the resulting value.

Step 3. Convergence check

i) Compute ${}_{(n)}\theta_p$ in (2.1) for all $p = 1, \ldots, P$ with ${}_{(n)}E_j$.

ii) Let

$$Z = \sum_{p=1}^{P} |{}_{(n)}\theta_p - {}_{(n-1)}\theta_p| / {}_{(n)}\theta_p.$$

iii) If $Z \leq \epsilon$, we adopt ${}_{(n)}\theta_p$ $(p = 1, \ldots, P)$ as approximate solutions to θ_p. Otherwise, add one to n and go to Step 2.

Even though we could not prove the convergence of the above iterative procedure, it converged in all of our numerical experiments shown in Subsection 2.2.

2.1.4 Blocking Probabilities in ABT/DT

Let $\boldsymbol{m}_j$ denote the state of link j:

$$\boldsymbol{m}_j = (m_{j,1}^{(1)}, \ldots, m_{j,H}^{(1)}, m_{j,2}^{(2)}, \ldots, m_{j,H}^{(2)}),$$

where $m_{j,h}^{(1)}$ $(1 \leq h \leq H)$ denotes the number of bursts being successfully transmitted on link j and having h hops, and $m_{j,h}^{(2)}$ $(h = 2, \ldots, H)$ denotes the number of bursts in transmission on link j, whose connection is in $R_j^{(h)}$, while being failed in transmission in one of the downstream links on the route. We define $h(\boldsymbol{m}_j)$ as the remaining amount of bandwidth given $\boldsymbol{m}_j$:

$$h(\boldsymbol{m}_j) = B_j - b\sum_{h=1}^{H} m_{j,h}^{(1)} - b\sum_{h=2}^{H} m_{j,h}^{(2)}.$$

Further we re-define S_j as

$$S_j = \{\boldsymbol{m}_j \mid h(\boldsymbol{m}_j) \leq B_j, m_{j,h}^{(1)} \geq 0\ (h = 1, \ldots, H), m_{j,h}^{(2)} \geq 0\ (h = 2, \ldots, H)\}, \qquad j = 1, \ldots,$$

Note that S_j denotes a feasible set of states of link j.

We now consider holding times of reservations from connection $p \in R_j^{(h)}$ ($h = 2, \ldots, H$) on link j, which fail in reservation in one of downstream links on the route. Note that the failure of the transmission is notified after time hD. Thus, the holding time is given by hD. On the other hand, when the connection p succeeds in reservation, the mean holding time μ_p^{-1} of the connection p in each link is given by

$$\mu_p^{-1} = (\mu b)^{-1} + h_p^* D,$$

where $h_p^* = |J^{(p)}|$ denotes the number of links on the route of connection p (i.e., the number of hops of connection p).

We re-define a $1 \times (2H-1)$ unit vector $\boldsymbol{e}_h$ ($h = 1, \ldots, 2H-1$) as

$$\boldsymbol{e}_h = (0, \ldots, 0, \underset{h\text{th}}{1}, 0, \ldots, 0).$$

The transition from state $\boldsymbol{m}_j$ to state $\boldsymbol{m}_j + \boldsymbol{e}_h$ ($h = 1, \ldots, H$) happens with rate $r_{j,h}^{(s)}$, where, for $h = 1, \ldots, H$,

$$r_{j,h}^{(s)} = \begin{cases} \displaystyle\sum_{p \in R_j^{(h)}} \lambda_p \prod_{k \in J_{j+}^{(p)}} (1 - E_k) \prod_{k \in J_{j-}^{(p)}} (1 - E_k), & \text{if } \boldsymbol{m}_j \in S_j, h(\boldsymbol{m}_j) > b, \\ 0, & \text{otherwise,} \end{cases} \tag{2.5}$$

where $H^{(h)}$ denotes the set of connections having h hops on their routes. Further the transition from state $\boldsymbol{m}_j$ to state $\boldsymbol{m}_j + \boldsymbol{e}_{H-1+h}$ ($h = 2, \ldots, H$) happens with rate $r_{j,h}^{(f)}$, where, for $h = 2, \ldots, H$,

$$r_{j,h}^{(f)} = \begin{cases} \displaystyle\sum_{p \in R_j^{(h)}} \lambda_p \prod_{k \in J_{j+}^{(p)}} (1 - E_k) \left[1 - \prod_{k \in J_{j-}^{(p)}} (1 - E_k) \right], & \text{if } \boldsymbol{m}_j \in S_j, h(\boldsymbol{m}_j) > b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Note again that all bursts are generated according to Poisson processes. Therefore we approximate the arrival process from each connection to link j by a Poisson process and we aggregate all arrivals into one Poisson stream. The traffic intensity ρ_j of the aggregated stream in link j is given by

$$\rho_j = \sum_{h=1}^{H} r_{j,h}^{(s)} \left\{ (\mu b)^{-1} + hD \right\} + \sum_{h=2}^{H} r_{j,h}^{(f)} hD.$$

Let M_j be

$$M_j = \sum_{h=1}^{H} m_{j,h}^{(1)} + \sum_{h=2}^{H} m_{j,h}^{(2)}$$

It then follows from the insensitivity property of $M/G/c/c$ that [51]

$$\Pr(M_j = k) = \frac{\rho_j^k/k!}{\sum_{i=0}^{K_j} \rho_j^i/i!}, \qquad k = 0, 1, \ldots, K_j,$$

where K_j denotes the maximum integer which is not greater than B_j/b. Thus the blocking probability E_j on link j is given by

$$E_j = \Pr(M_j = K_j) = \frac{\rho_j^{K_j}/K_j!}{\sum_{i=0}^{K_j} \rho_j^i/i!}, \tag{2.7}$$

We now provide an iterative procedure to obtain the blocking probability E_j. In what follows, for any symbol X, we denote the value of X in the nth iteration by ${}_{(n)}X$.

Step 1. Initial input: for all $p = 1, \ldots, P$ and all $j = 1, \ldots, J$,

i) Let ${}_{(0)}E_j = 0$

ii) Let ${}_{(0)}\theta_p = \lambda_p/\mu$.

iii) Set a nonnegative small value to ϵ (e.g., $\epsilon = 10^{-3}$ for graphical representations).

iv) Let $n = 1$.

Step 2. The nth iteration:

i) Compute the arrival rates $r_{j,h}^{(s)}$ $(h = 1\ldots., H)$ in (2.5) and the arrival rates $r_{j,h}^{(f)}$ $(h = 2\ldots., H)$ in (2.6) for all $j = 1, \ldots, J$ with ${}_{(n-1)}E_j$.

ii) Compute the right hand side of (2.7) and let ${}_{(n)}E_j$ be the resulting value.

Step 3. Convergence check

i) Compute ${}_{(n)}\theta_p$ in (2.1) for all $p = 1, \ldots, P$ with ${}_{(n)}E_j$.

ii) Let

$$Z = \sum_{p=1}^{P} |{}_{(n)}\theta_p - {}_{(n-1)}\theta_p| / {}_{(n)}\theta_p.$$

iii) If $Z \leq \epsilon$, we adopt ${}_{(n)}\theta_p$ $(p = 1, \ldots, P)$ as approximate solutions to θ_p. Otherwise, add one to n and go to Step 2.

Even though we could not prove the convergence of the above iterative procedure, it converged in all of our numerical experiments shown in Subsection 2.2.

2.2 Performance Comparisons of ABT/DT and IT

2.2.1 Network Model

In this subsection, we compare performances of ABT/DT and IT based on the approximate analysis presented in the previous subsection. We use the tandem network model with two links throughout this section (see Fig. 2.1) except Subsection 2.2.5, where a more general network model is treated to show that observations made in Subsections 2.2.3 and 2.2.4 are also applicable to more general network topologies.

As shown in Fig. 2.1, two–hop connection C1 contends with one–hop connection C2 for link L1, and does with another one–hop connection C3 for link L2. The capacities of two ATM links, B_j $(j =$ L1, L2), are identically set to 150 Mbps. The generation rate of bursts at sources are identically set to λ_0, i.e., $\lambda_p = \lambda_0$ $(p =$ C1, C2, C3). The mean burst length, $1/\mu$, is 5 Kbits, corresponding to 33 μsec on 150 Mbps link. The propagation delays of links L1 and L2 are varied from 1 μsec to 1 msec.

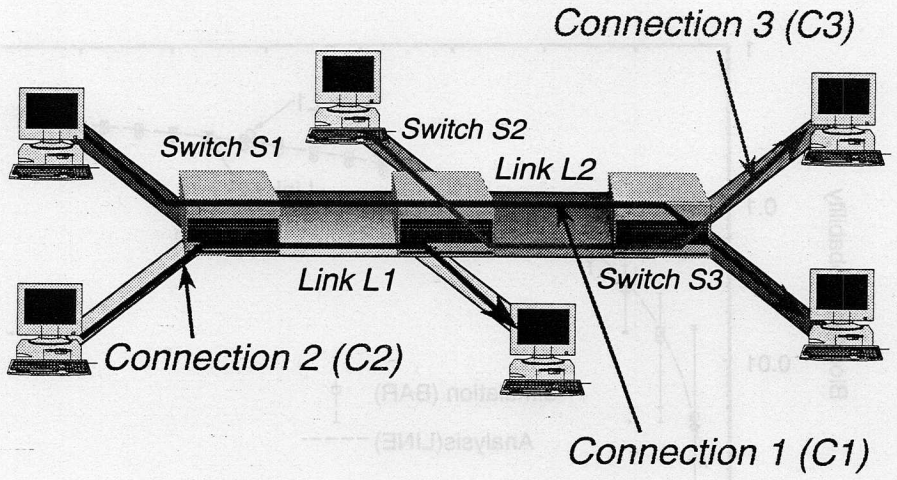


Figure 2.1: Network Model

2.2.2 Accuracy of Our Approximate Analysis

We first assess the accuracy of our approximate analysis by comparing with simulation. In Figs. 2.2 and 2.3, we compare blocking probability values of ABT/IT with 1 μsec and 1 msec propagation delays, respectively. The requested bandwidth b is set to be 75 Mbps. Simulation results are shown with 95% confidence intervals. We can observe good agreements between analysis and simulation results. The corresponding results for ABT/DT are also plotted in Figs. 2.4 and 2.5. Excellent accuracies can also be observed in those figures.

2.2.3 Effect of Offered Load on Throughput

Recalling that throughputs can directly be calculated from the blocking probability (see, eq. (1)), we first compare throughputs of ABT/DT and IT against the offered load (λ_0/μ) in Fig. 2.6. The propagation delay of each link is set to be 1 μsec, and the requesting bandwidth b is 50 Mbps. In the figure, results for three connections C1, C2 and C3 are displayed. Since connection C1 is two–hop connection, its throughput is degraded as the offered load becomes high. The difference between ABT/DT and IT cannot be observed in the case of the short propagation delay.

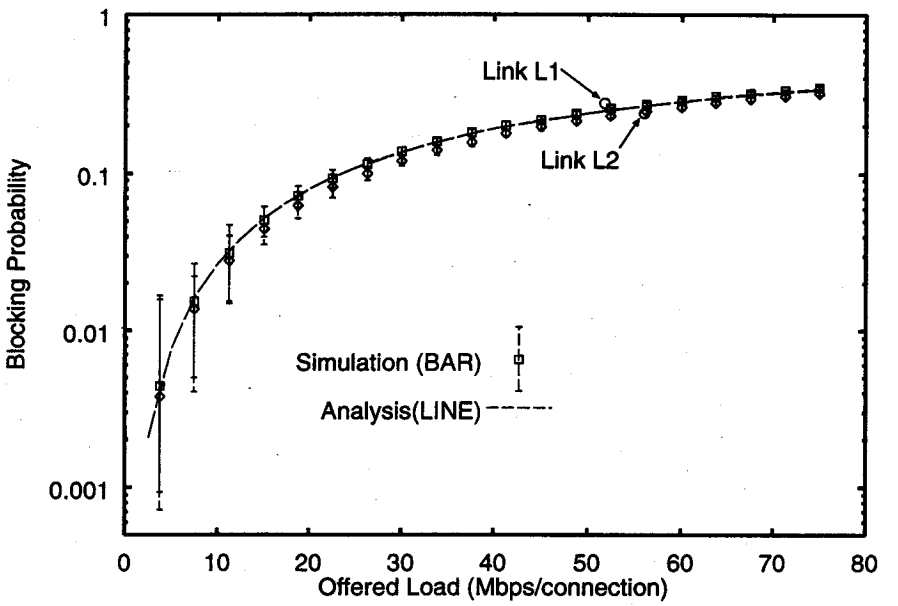


Figure 2.2: Comparisons of Analysis and Simulation (ABT/IT, Propagation Delay = 1 μsec)

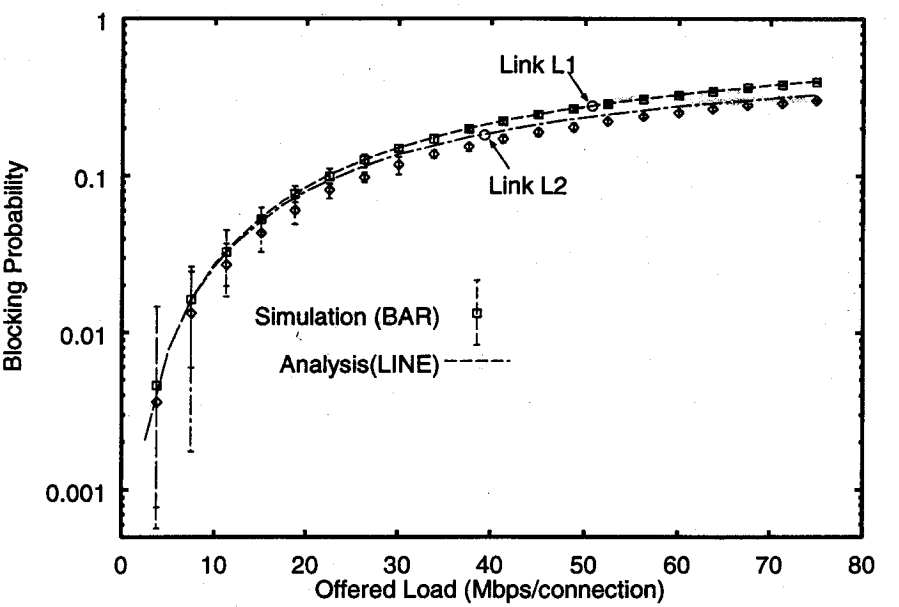


Figure 2.3: Comparisons of Analysis and Simulation (ABT/IT, Propagation Delay = 1 msec)

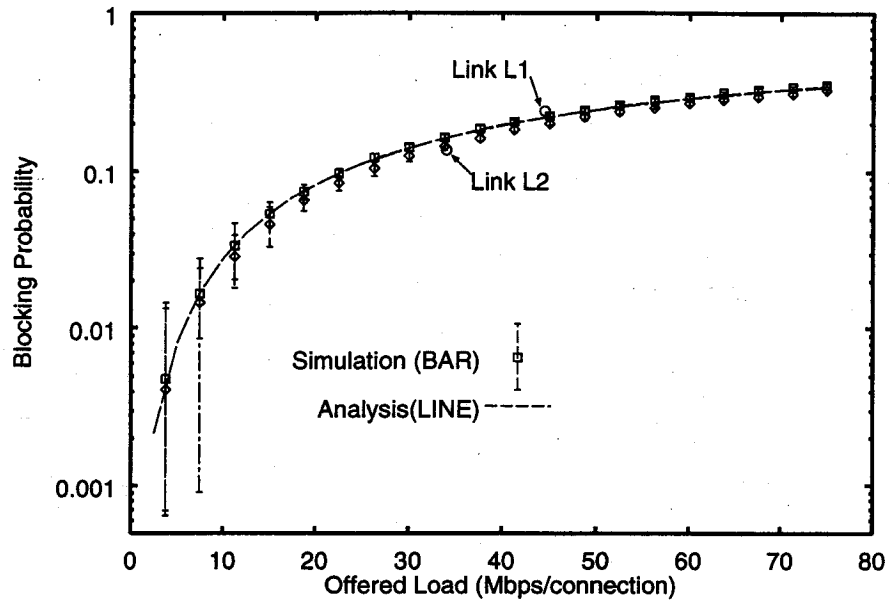


Figure 2.4: Comparisons of Analysis and Simulation (ABT/DT, Propagation Delay = 1 μsec)

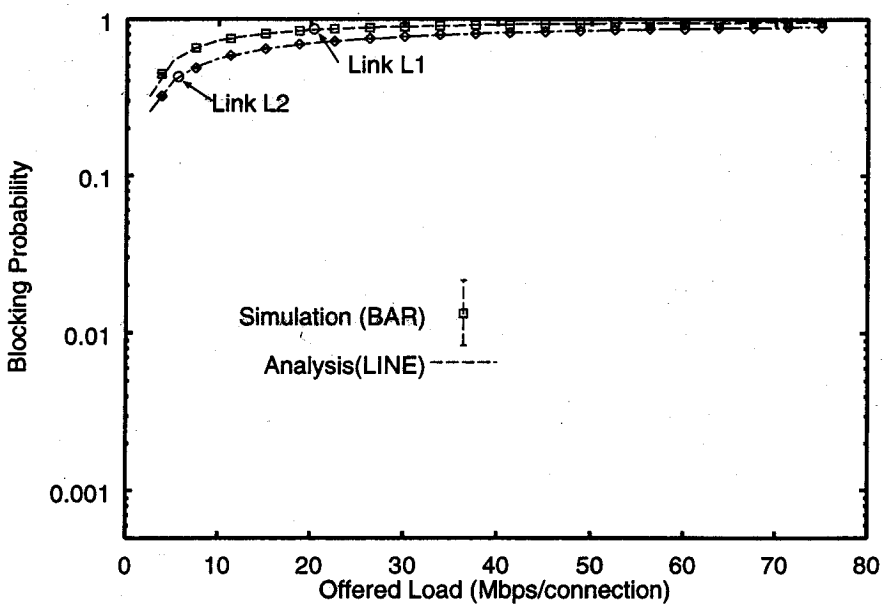


Figure 2.5: Comparisons of Analysis and Simulation (ABT/DT, Propagation Delay = 1 msec)

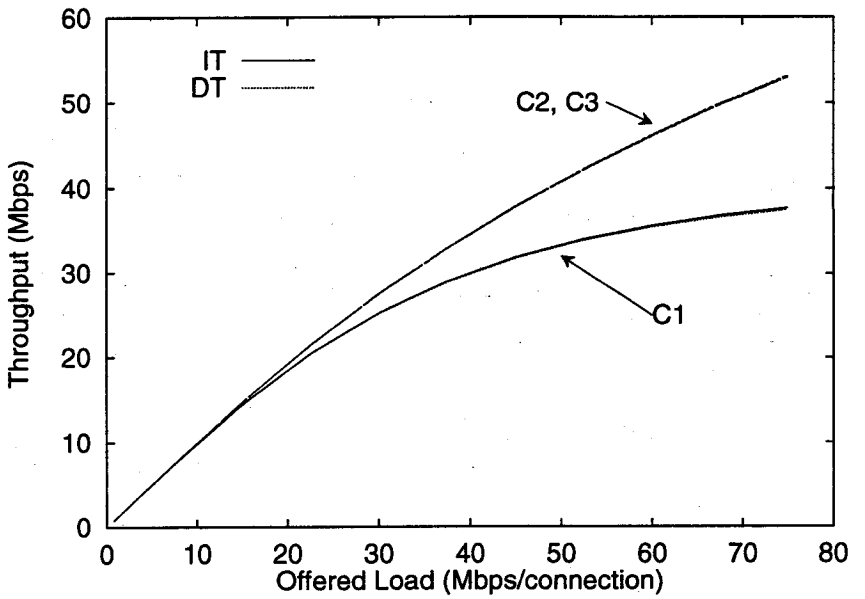


Figure 2.6: Throughput vs. Offered Load (Propagation Delay = 1 μsec)

The difference becomes significant when the propagation delay is set to be 1 msec as shown in Fig. 2.7. In the case of ABT/DT, the large propagation delay leads to the larger blocking probability of link L1 and henceforth lower throughput of connection C1. The offered load on link L2 then becomes smaller than that of link L1. It is the reason that the throughput of connection C3 is less degraded than those of other connections.

2.2.4 Effect of Propagation Delay on Throughput

The effect of the propagation delay is illustrated more clearly in Fig. 2.8, where the throughput of connection C1 against the propagation delay is shown. In obtaining the figure, the requesting bandwidth b is set to be 50 Mbps, and the offered load on the link is varied as ρ = 0.9 (135 Mbps), 0.5 (75 Mbps) and 0.1 (15 Mbps). We can observe the throughput of ABT/DT is degraded suddenly when the propagation delay becomes around 0.1 msec, and reaches almost zero when the propagation delay is 10 msec. Those values of propagation delays correspond to 20 Km and 2000 Km

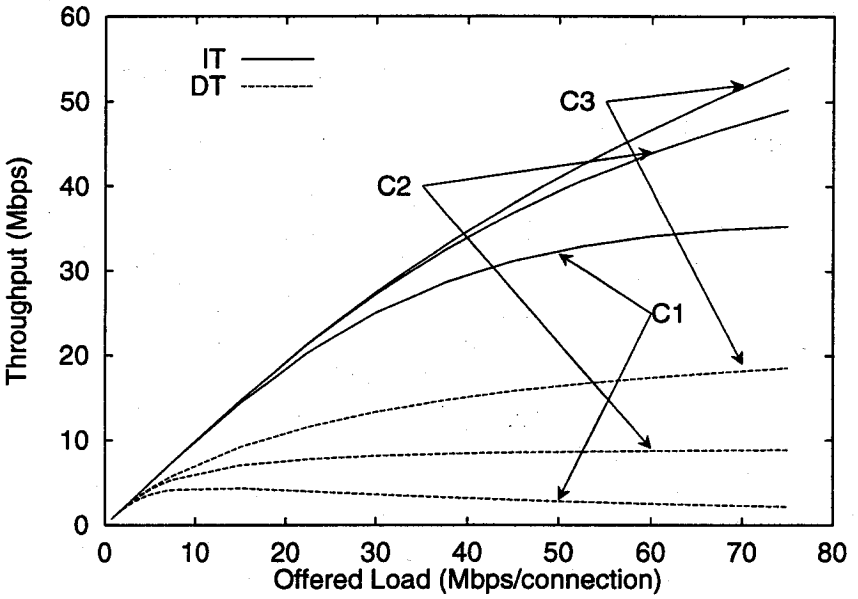


Figure 2.7: Throughput vs. Offered Load (Propagation Delay = 1 msec)

long, respectively. Namely, ABT/DT is not applicable to metropolitan and wide area networks. The throughput of ABT/IT is also degraded by the large propagation delay, but the influence is very small because the bandwidth reservation time is not affected by the propagation delays.

We next illustrate the effect of the requesting bandwidth, b, on the throughput of connection C1 in Fig. 2.9. The requesting bandwidth b is varied as 150 Mbps, 75 Mbps, 50 Mbps and 37.5 Mbps, and the offered load ρ is 0.9. From the figure, we can verify that performance tendencies of ABT/DT and IT are not affected by the amount of the requesting bandwidth. Same observation can also be made for other connections C2 and C3 as shown in Fig. 2.10.

2.2.5 Application to the Random Network Model

In the previous subsections, we have treated a rather simple model depicted in Fig. 2.1. In this subsection, we examine more general network topologies. For this purpose, we generate random networks [52] with twenty nodes in 5×4 matrix. The

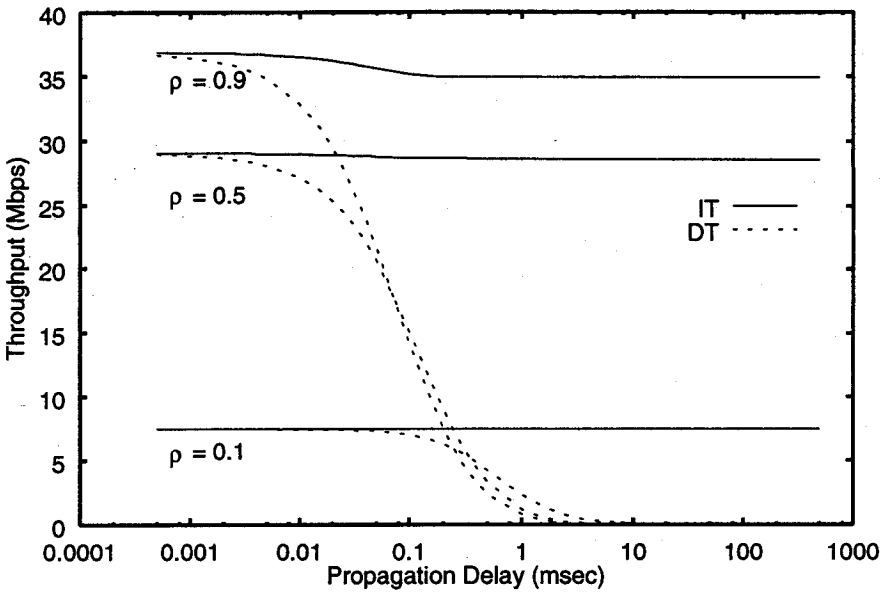


Figure 2.8: Throughput Comparisons of ABT/DT and IT dependent on Propagation Delay (Connection C1)

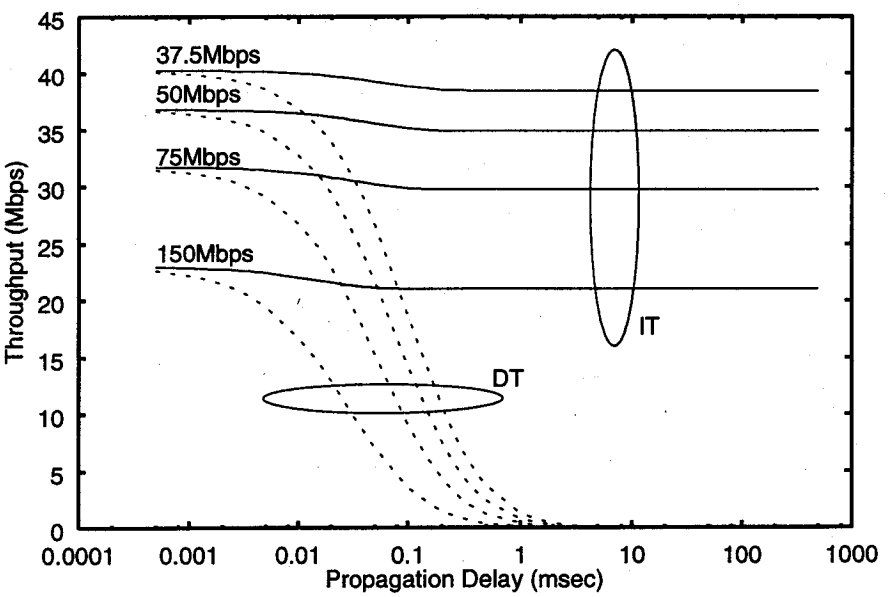


Figure 2.9: Throughput Comparisons of Bandwidths (C1, $\rho = 0.9$)

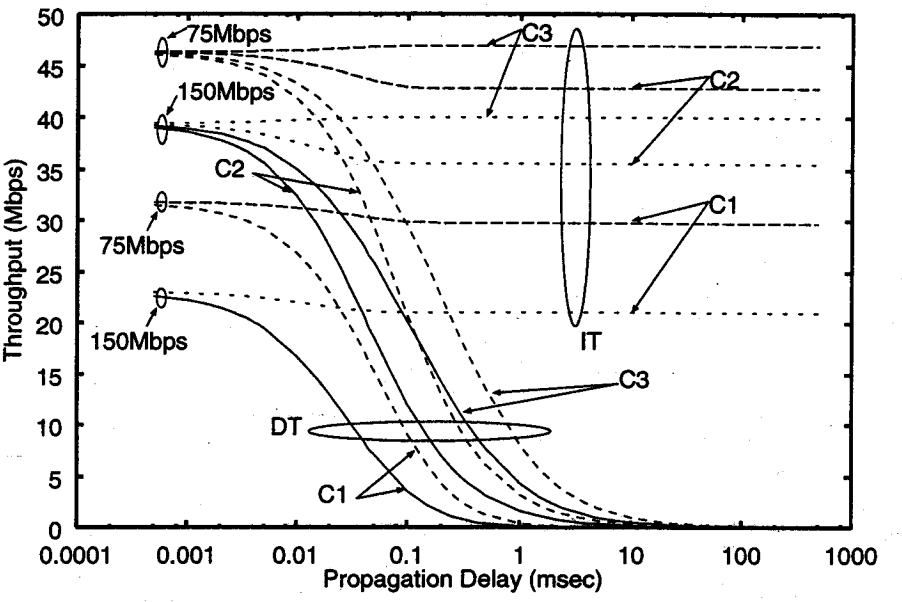


Figure 2.10: Throughput Comparisons of Connections ($\rho = 0.9$)

link between two nodes is generated randomly. The connection between every two terminals is established with the shortest–path. If there are multiple shortest–paths with identical length, the path for connection establishment is chosen randomly.

Figure 2.11 compares throughputs of ABT/DT and IT against the offered load. As shown in the figure, we can observe the similar tendency as in Subsection 2.2.3, the performance of ABT/DT degrades dramatically by the larger propagation delays while that of IT does not. Throughputs of IT and DT against the propagation delay are compared in Fig. 2.12. We again observe the same tendency as in Subsection 2.2.4 even for random networks.

Next, throughputs of DT and IT are compared in Fig. 2.13 for two values of propagation delays; 1 μsec and 1 msec. The offered load of each connection is identically set to be 0.75 Mbps. In the figure, mean throughput dependent on the number of hops is plotted. As shown in the figure, the performance of ABT/DT is low and reaches almost zero as the number of hops becomes large while ABT/IT gives good performance independent of the number of hops. However, it is not true when the traffic load becomes high. As shown in Fig. 2.14, the throughput is suddenly de-

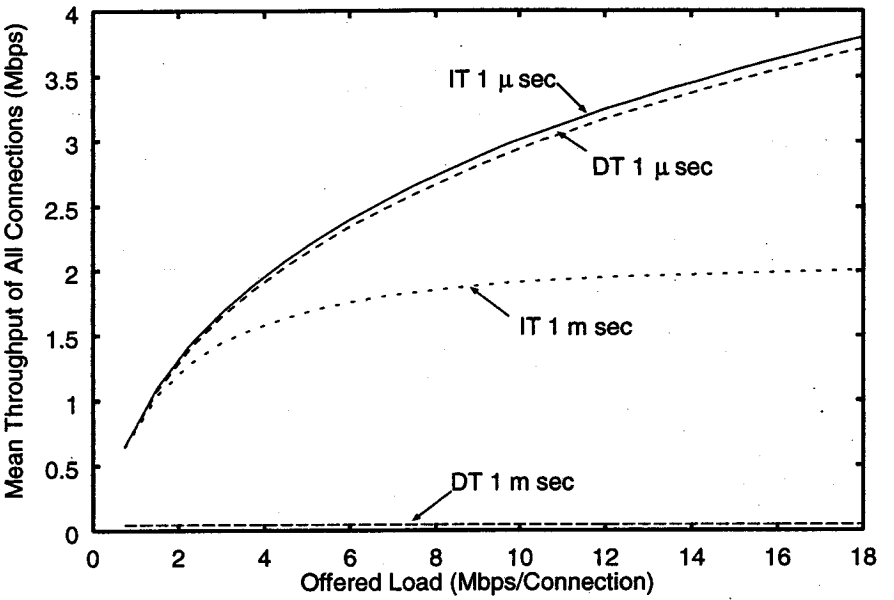


Figure 2.11: Effect of Offered Load in Random Network (Offered Load = 7.5 Mbps)

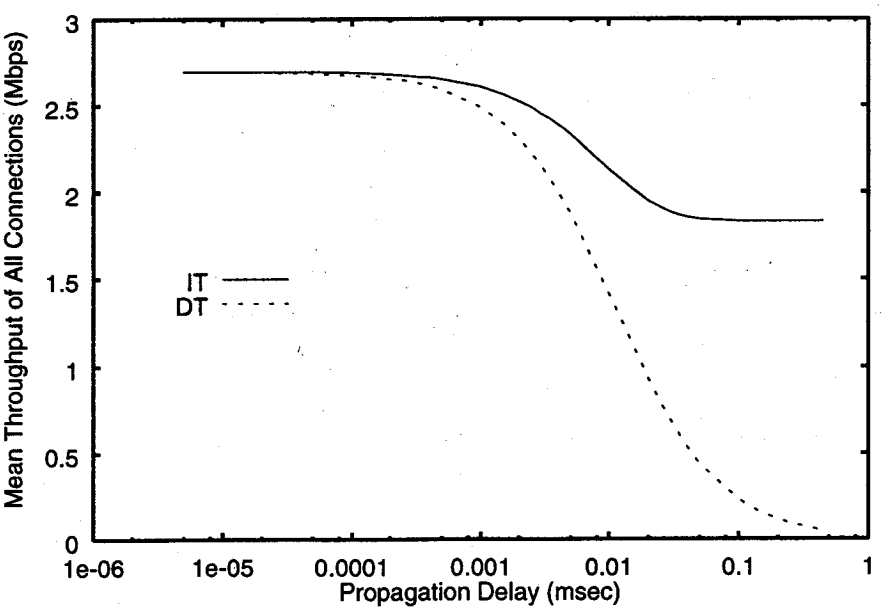


Figure 2.12: Effect of Propagation Delay in Random Network (Offered Load = 7.5 Mbps)

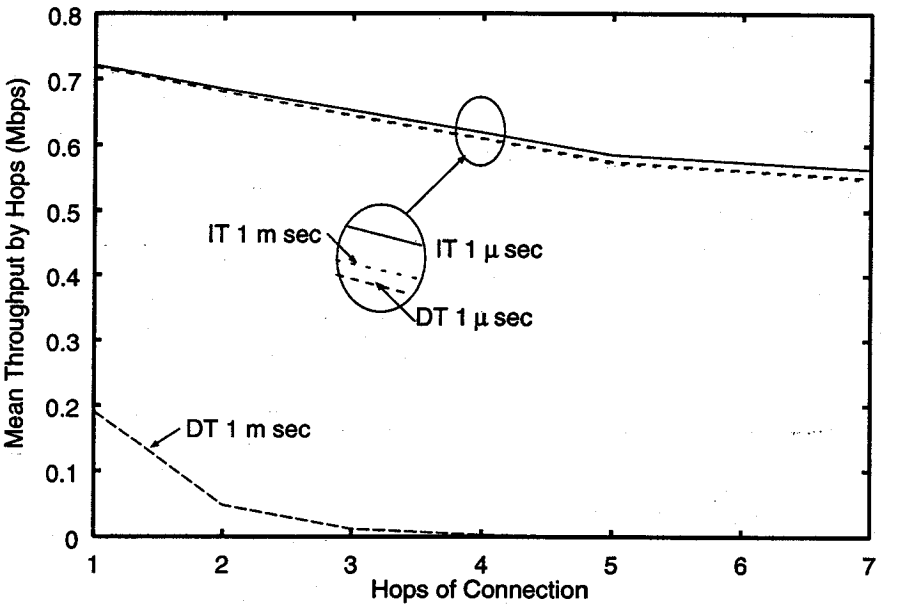


Figure 2.13: Effect of the Number of Hops on Throughput (Offered Load = 0.75 Mbps)

creased for connections with the larger number of hops even in ABT/IT.

2.3 Effects of Flexible Bandwidth Reservation Mechanisms

2.3.1 Effects of Bandwidth Negotiation in ABT/DT

In this subsection, we first consider the effect of the bandwidth negotiation mechanism which is only applicable to ABT/DT. By this mechanism, the blocking probability is expected to be decreased by accepting the reduced amount of bandwidth. More specifically, we consider the following bandwidth negotiation mechanism. Each source requests the bandwidth with an initial value, b, by using the forward RM cell. If the requested bandwidth b is available on the link, the switch simply accepts the request. On the other hand, if the available bandwidth of the link is smaller than that value, the switch checks whether the half of the requested bandwidth is

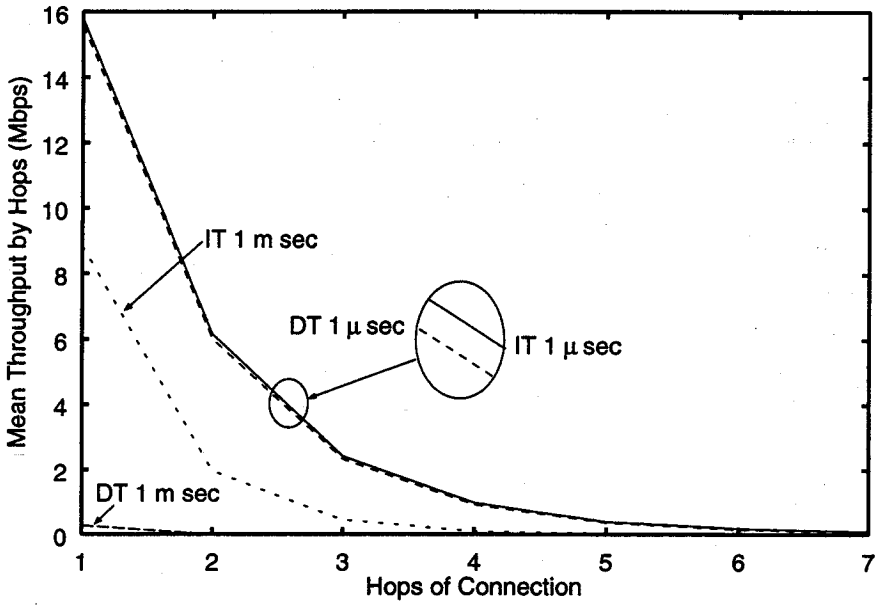


Figure 2.14: Effect of the Number of Hops on Throughput (Offered Load = 37.5 Mbps)

available or not. If it is available, the switch reserves the bandwidth of $b/2$ and overwrites it in the RM cell. If not, on the other hand, the switch again checks whether another half ($b/4$) is available or not. In this way, the bandwidth is reduced until the available bandwidth is found. When the switch receives the backward RM cell, it adjusts the reserved bandwidth to the one specified in the RM cell. Recall that such a negotiation cannot be implemented in ABT/IT since the burst is transmitted immediately following the RM cell. For evaluating the bandwidth negotiation mechanism, we use the model in Fig. 2.1. The initial requesting bandwidth b is set to be 150 Mbps. In simulation, we set the minimum bandwidth to be 150/16 Mbps. Namely, if the available bandwidth on the link is less than 150/16 Mbps, the reservation request is rejected. It prevents the reserved bandwidth from being much less than the requesting bandwidth.

We first examine how the throughput of ABT/DT can be improved by introducing the bandwidth negotiation. Figure 2.15 shows the case of the short propagation delay, 1 μsec. As shown in the figure, the throughput of ABT/DT with bandwidth

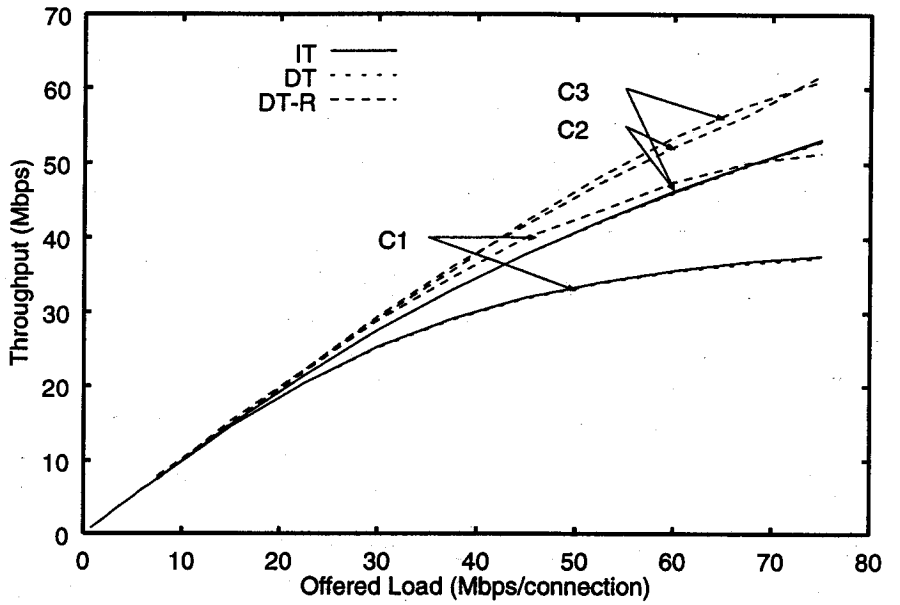


Figure 2.15: Effect of Bandwidth Negotiation in ABT/DT (Propagation Delay = 1 μsec)

negotiation (labeled by "DT–R" in the figure) can be much improved. It becomes even larger than that of ABT/IT in which the bandwidth negotiation cannot be implemented. However, improvement is limited when the propagation delay becomes large as shown in Fig. 2.16, where the propagation delay is set to be 1 msec.

Improved performance in previous figures was obtained by reducing the reserved bandwidth. It implies that the burst transmission time becomes longer. We next use a *power* index, which is defined as a ratio of throughput to transmission delay. Here, the transmission delay is a time duration from burst generation at the source to its successfully reception at the destination. The results are plotted in Figs. 2.17 and 2.18 against the offered load for two values of propagation delays, 1 μsec and 1 msec, respectively. Figure 2.17 shows that the performance of ABT/DT with bandwidth negotiation is not good when we are concerned with the burst transmission delay. In other words, the effect of bandwidth negotiation in ABT/DT is meaningful in LAN environment if our main concern is only throughput. Otherwise, ABT/IT still gives better performance even in such a circumstance.

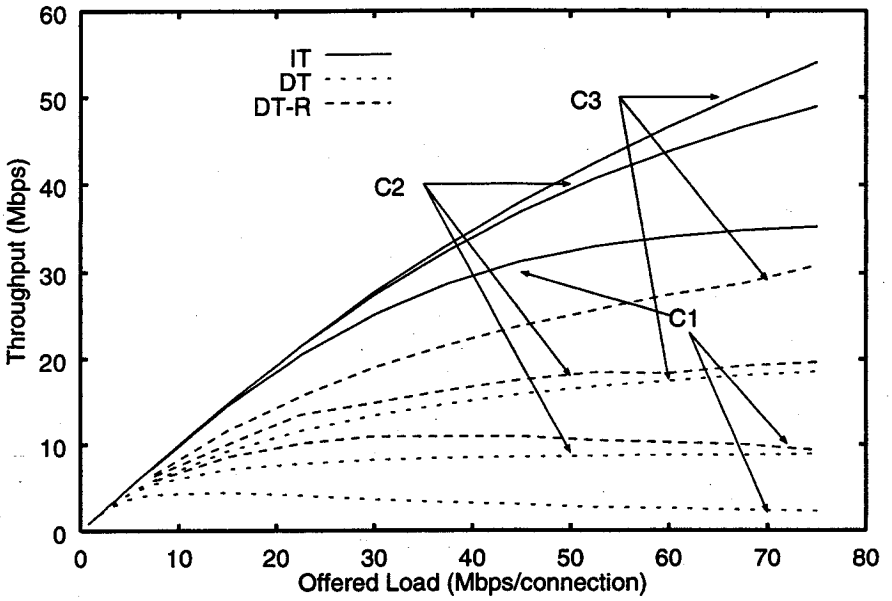


Figure 2.16: Effect of Bandwidth Negotiation in ABT/DT (Propagation Delay = 1 msec)

In the above experiments, we have assumed that the burst without successfully bandwidth reservation is lost. In the next subsection, we will investigate the case where the reservation request is repeated until it is successfully admitted.

2.3.2 Performance Comparisons of ABT/IT and DT with Backoff Methods

We last compare the performance by taking account of the backoff algorithm in both of ABT/DT and IT protocols. By the backoff, we mean that if the bandwidth reservation is rejected, the source waits during some time period (backoff interval) to retry reservation later. Since the reservation failure is an indication of congestion on some link of the route, the bandwidth reduction after the backoff could lead to the acceptance of the request. In [19], the authors compare several bandwidth reduction methods in ABT/DT, and concluded that the appropriate method is to reduce the requesting bandwidth to half after the reservation failure. In the current section, we

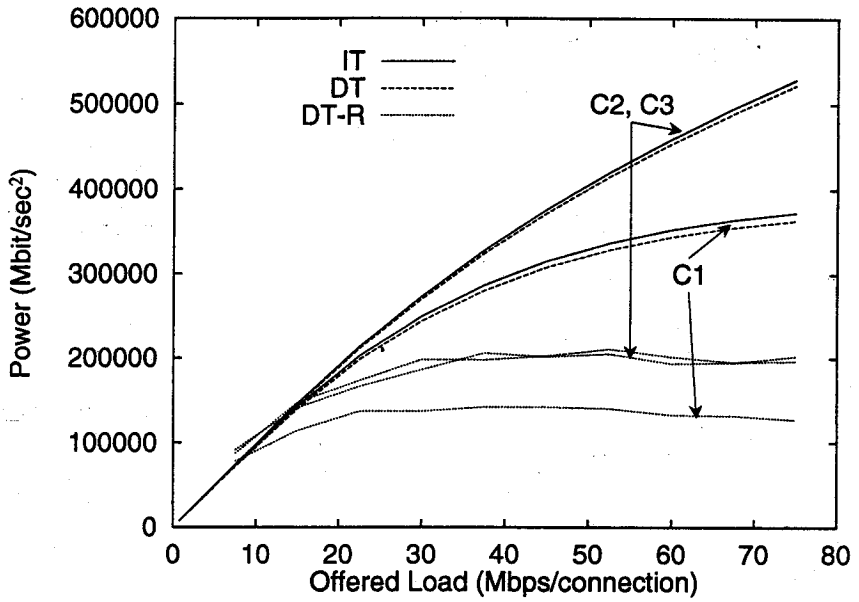


Figure 2.17: Effect of Bandwidth Negotiation in ABT/DT on Power (Propagation Delay = 1 μsec)

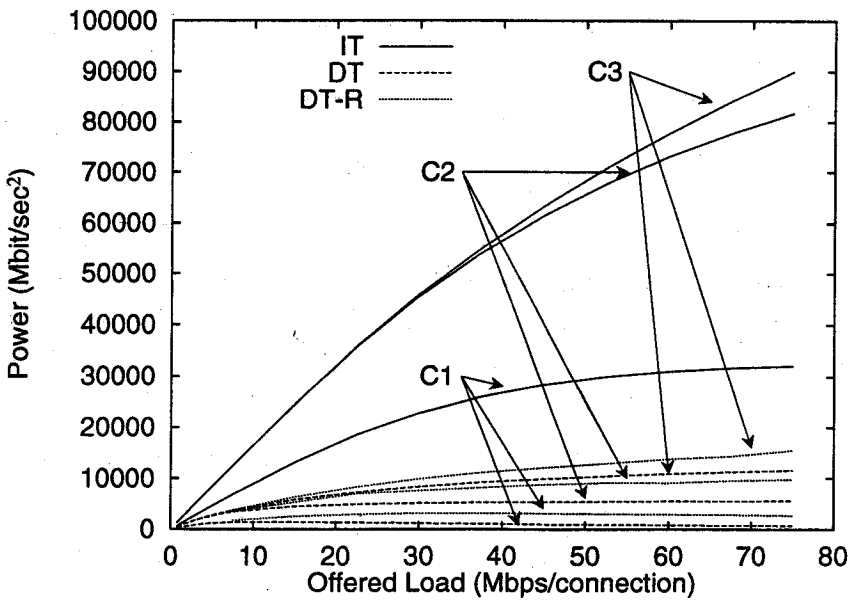


Figure 2.18: Effect of Bandwidth Negotiation in ABT/DT on Power (Propagation Delay = 1 msec)

also consider such a bandwidth reduction method to compare ABT/DT and IT. In simulation, the following five methods are compared.

Method M1: ABT/DT in which the requesting bandwidth is always fixed even after the backoff.

Method M2: ABT/IT with fixed bandwidth for reservation as in Method M1.

Method M3: ABT/DT in which the requesting bandwidth is reduced to half after each backoff.

Method M4: ABT/IT, the bandwidth reduction method is same as Method M3.

Method M5: In the above four methods, the dynamic bandwidth negotiation mechanism in the previous subsection is not considered. In this Method M5, the requesting bandwidth is fixed even after the backoff, but the bandwidth negotiation presented in Subsection 4.1 is allowed for ABT/DT.

For the backoff time, we assume that it is distributed exponentially and its mean is set to be 300 μsec.

We first compare the burst transmission delays of Methods M1 and M2 in Fig. 2.19. The propagation delay is set to be short, 1 μsec. As shown in the figure, the difference of transmission delays between ABT/IT and DT is small. However, the burst transmission delay of ABT/DT becomes worse dramatically by the long propagation delay. As shown in Fig. 2.20 for the case of 1 msec propagation delay, it can easily be conjectured from the previous results of throughputs (Figs. 2.8 through 2.10).

We now investigate the effect of bandwidth reduction after the backoff. Figure 2.22 compares Methods M2 and M4, i.e., ABT/IT with and without reduction of the requesting bandwidth. The propagation delay is 1 μsec. Initial bandwidths of Method M2 are varied as 150 Mbps, 75 Mbps, 37.5 Mbps and 18.75 Mbps. For Method M4, we only show the case where the initial bandwidth is set to be 75 Mbps. It can be observed that the performance of Method M4 is better than that of M2 in a

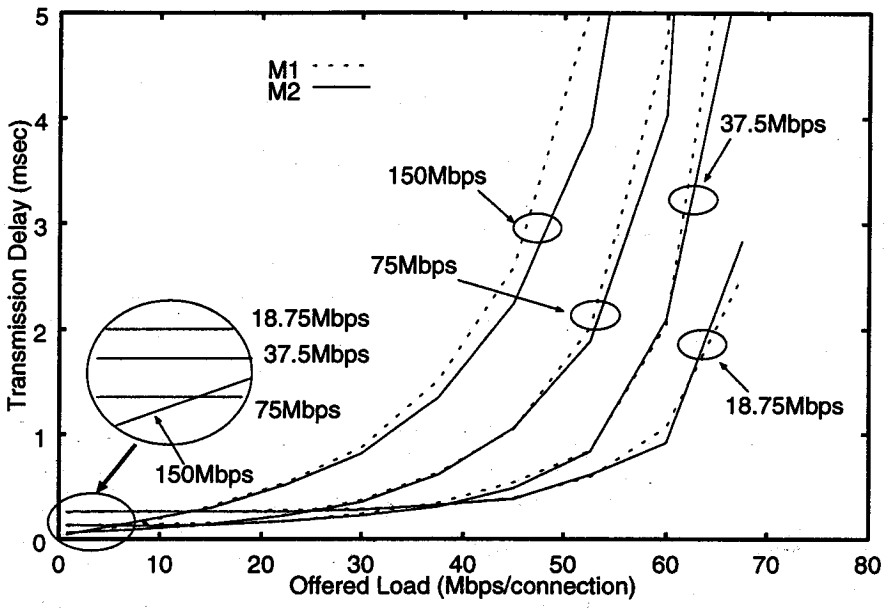


Figure 2.19: Transmission Delay dependent on Offered Load (Methods M1 and M2, Propagation Delay = 1 μsec)

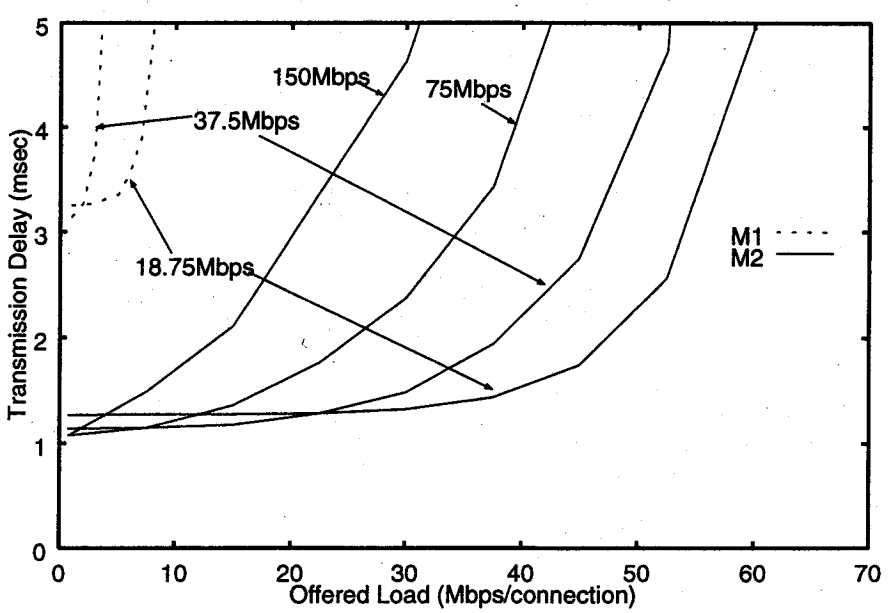


Figure 2.20: Transmission Delay dependent on Offered Load (Methods M1 and M2, Propagation Delay = 1 msec)

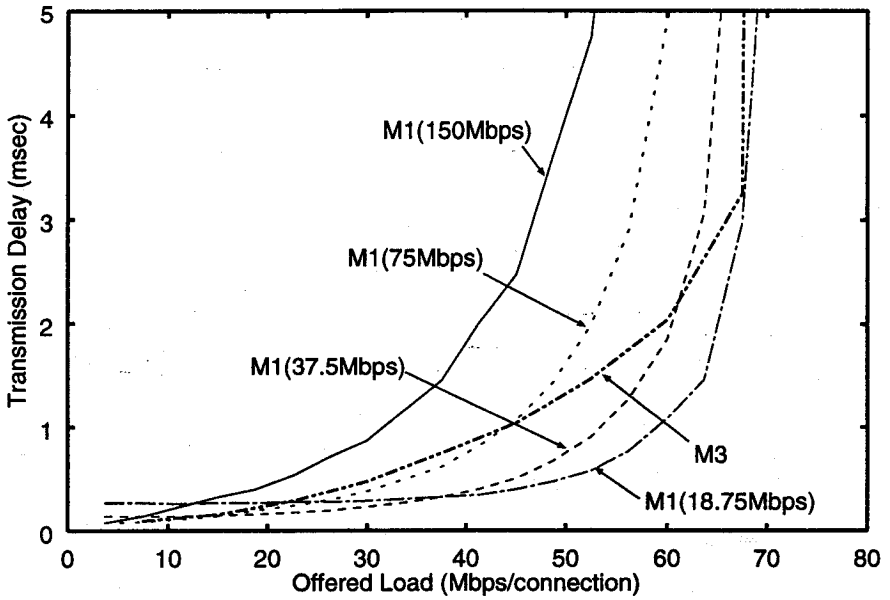


Figure 2.21: Transmission Delay Comparisons of M1 and M3 (Propagation Delay = 1 μsec)

sense that it can offer fairly good performance independent of the traffic load. It is because the reduced bandwidth of Method M4 leads to avoid the repeated backoffs as we expect. We can observe a similar result in the case of ABT/DT as shown in Fig. 2.21, where we compare Methods M1 and M3. In the case of long propagation delay, the similar results can be observed as shown in Fig. 2.23, where the propagation delay is set to be 1 msec.

We next compare five methods in Figs. 2.24 and 2.25 with 1 μsec and 1 msec propagation delays, respectively. As can be found in Fig. 2.24, Method M5 gives best performance when the propagation delay is small. It is because in Method M5, the reservation is admitted even when a small amount of the bandwidth is available on the link, which can avoid backoffs. However, Method M4 (IT with bandwidth reduction) is most effective in the case of the long propagation delay because the overhead of ABT/DT introduced by the long propagation delay cannot be overcome even when the dynamic bandwidth negotiation mechanism is introduced.

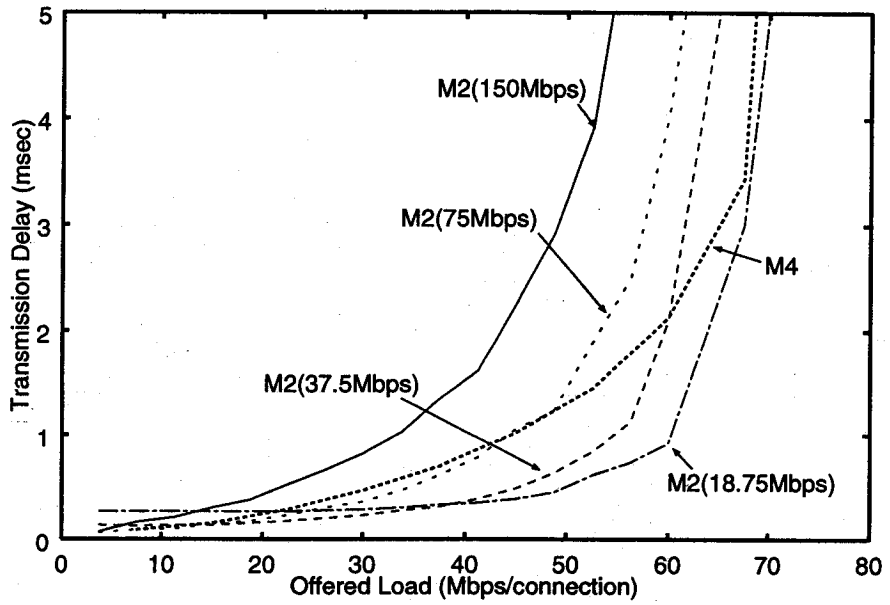


Figure 2.22: Transmission Delay Comparisons of M2 and M4 (Propagation Delay = 1 μsec)

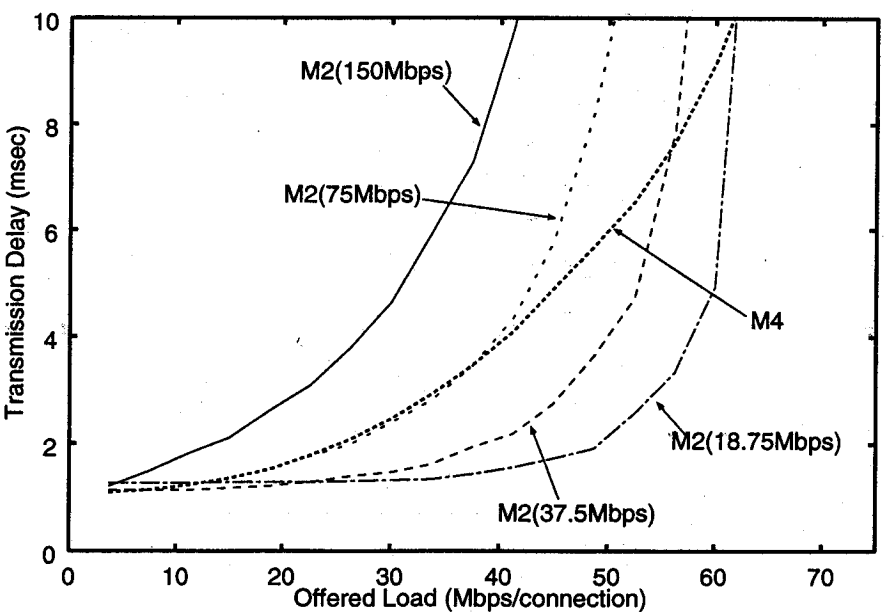


Figure 2.23: Transmission Delay Comparisons of M2 and M4 (Propagation Delay = 1 msec)

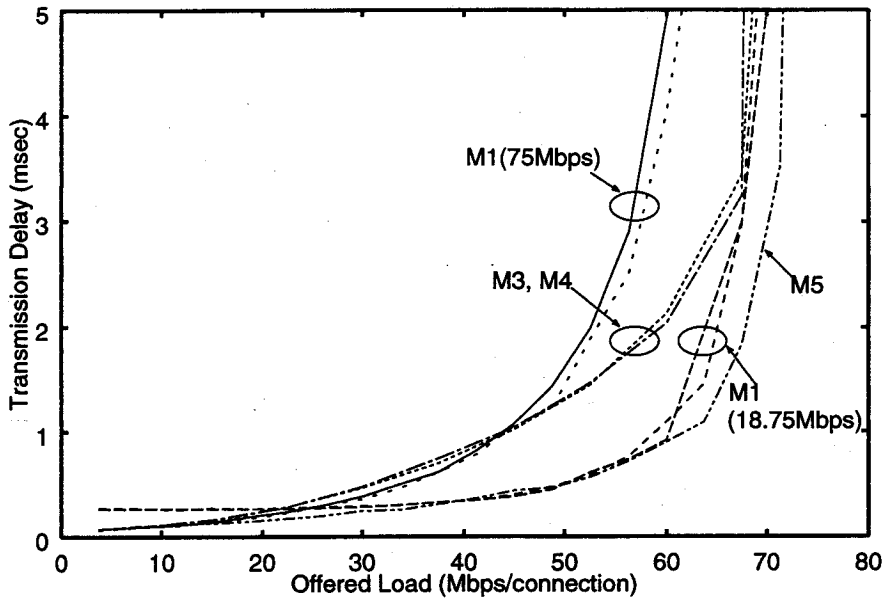


Figure 2.24: Transmission Delay Comparisons of Five Methods (Propagation Delay = 1 μsec)

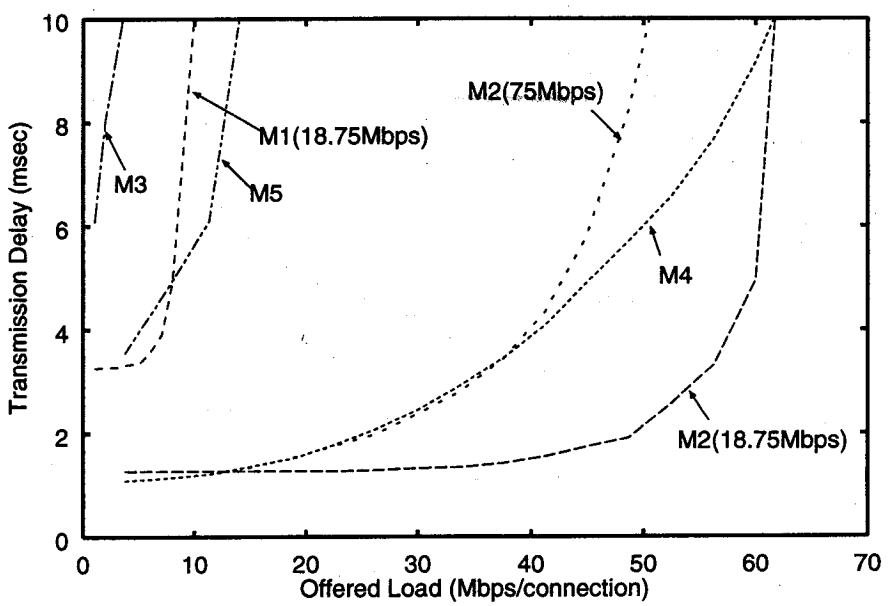


Figure 2.25: Transmission Delay Comparisons of Five Methods (Propagation Delay = 1 msec)

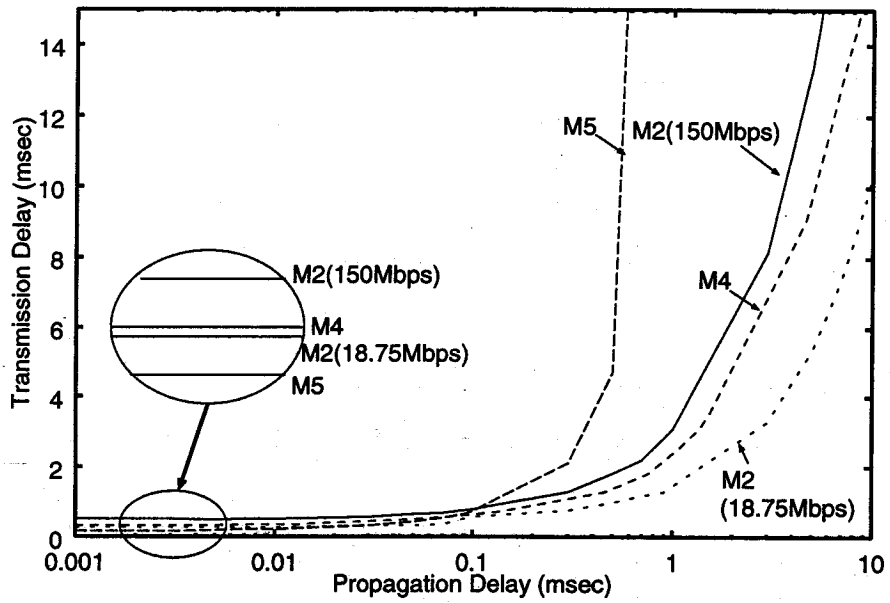


Figure 2.26: Transmission Delay Comparisons of M2, M4 and M5 dependent on Propagation Delay (Offered Load = 22.5 Mbps/Connection)

Last, the burst transmission delay comparisons of Methods M2, M4 and M5 dependent on the propagation delay are shown in Fig. 2.26. Method M5 outperforms other two methods if the propagation delay is small. However, as the propagation delay becomes longer than burst transmission times, the performance of Method M5 becomes worst suddenly. The same tendency was also observed in Subsection 2.2.4.

From the above experiments, we can see that Method M2 with a small reservation bandwidth exhibits a good throughput. However, it poses large transmission delays even when the traffic load is low. On the other hand, Method M4 (IT with bandwidth reduction) fairly gives a good performance in terms of both throughput and the transmission delay in all parameter regions we have tried.

2.4 Concluding Remarks

In this section, we have first investigated the basic performance of ABT/IT and DT. Then, we have shown that ABT/IT is robust in the sense that its performance is not

heavily affected by the propagation delay. On the other hand, ABT/DT is quite sensitive to the propagation delay. We next considered the performance improvement by the bandwidth negotiation mechanism which is only applicable to the ABT/DT protocol. Simulation results have shown that it is effective in the short propagation delay case if our concern is throughput, but the burst transmission delay is still larger than that of ABT/DT. We have also investigated effects of backoff methods to compare the burst transmission delays, and have observed similar tendencies as in the above cases. In the case of the short propagation delay, ABT/DT with bandwidth negotiation is most effective. When the propagation delay becomes large, on the other hand, ABT/IT with reduced bandwidth mechanism outperforms other methods.

Chapter 3

Performance Improvement of ABT Protocols with Combined Bandwidth/Buffer Reservation

In the previous chapter ABT/IT has been shown to be robust in that its performance is not much affected by the propagation delay when compared with ABT/DT. However, when the traffic load becomes heavy and/or when the number of hop counts of the connection becomes large, the throughput of ABT/IT is drastically decreased. We propose a new protocol, *Buffered ABT/IT*, which makes reservation on the buffer as well as the bandwidth. The approximate analysis method is then developed for buffered ABT/IT. Through numerical examples, we show that it can much improve the performance even in the above conditions by comparing with the exiting ABT protocols.

3.1 Algorithms of Buffered–ABT/IT

In this section, we describe the buffered–ABT/IT, which reserves the bandwidth and buffer at intermediate switches for burst transmission. To describe the algo-

rithm, we first introduce some notations (see also Fig. 3.1). The link bandwidth of the switch per each output port is denoted by B. The switch has a buffer capacity of F at each output port, and we assume that the switch can manage the buffer per connection. This is necessary because the burst is temporarily stored in the buffer if the bandwidth is not available. The switch maintains the usage status of the buffer and link for each output port. The amount of the available buffer, which is not being reserved by any burst, is denoted by F_a. The available link bandwidth is denoted by B_a. Furthermore, the switch has to maintain a table of the bursts temporarily stored in the buffer. Each entry of the table contains (1) the pointer to the first cell of the burst stored in the buffer, (2) the expected time at which the switch will start transmission of the burst stored in the buffer, (3) the amount of the bandwidth requested by the burst, b, and (4) the length of the burst, l. The third and fourth quantities are extracted from the forward RM cell when the corresponding burst is decided to be stored in the buffer. That is, the information on the burst length (l) should be specified in the RM cell in our protocol in addition to the requesting bandwidth (b) as in the original ABT protocol. In our model, we assume that the switch can serve multiple bursts simultaneously with a rate–based scheduling algorithm, and overheads of the processing time on the switch are negligible. With these information, the switch can serve the burst without any buffers when enough bandwidth is available on the link.

We now describe the algorithm for the buffered–ABT/IT. When the burst arrives at the source terminal, the forward RM cell with the requesting bandwidth (b) and the length of the burst (l) is sent to the destination along the predefined route, followed by the burst transmission according to the original ABT/IT protocol. Each switch receiving the RM cell examines the available bandwidth B_a. If a sufficient amount of the bandwidth is available on the link (i.e., $B_a \geq b$), it reserves the bandwidth by setting $B_a \leftarrow B_a - b$, and forwards the RM cell to the next switch in the downstream. Thus, the operation is identical to the original ABT in this case. The different treatment is performed when the requested bandwidth b is not available.

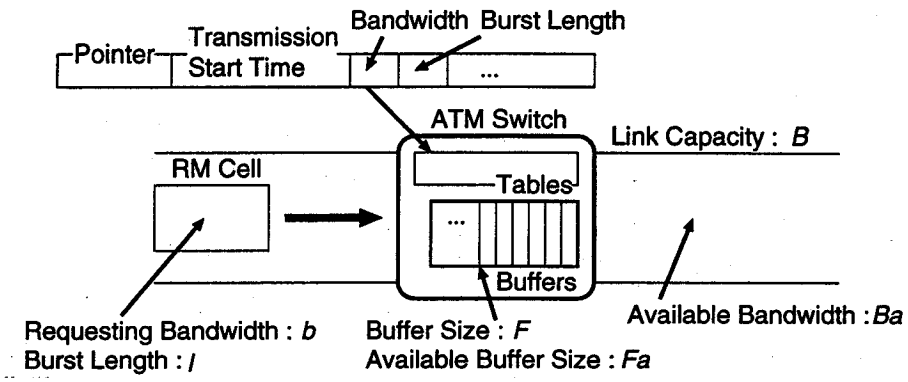


Figure 3.1: Switch Model

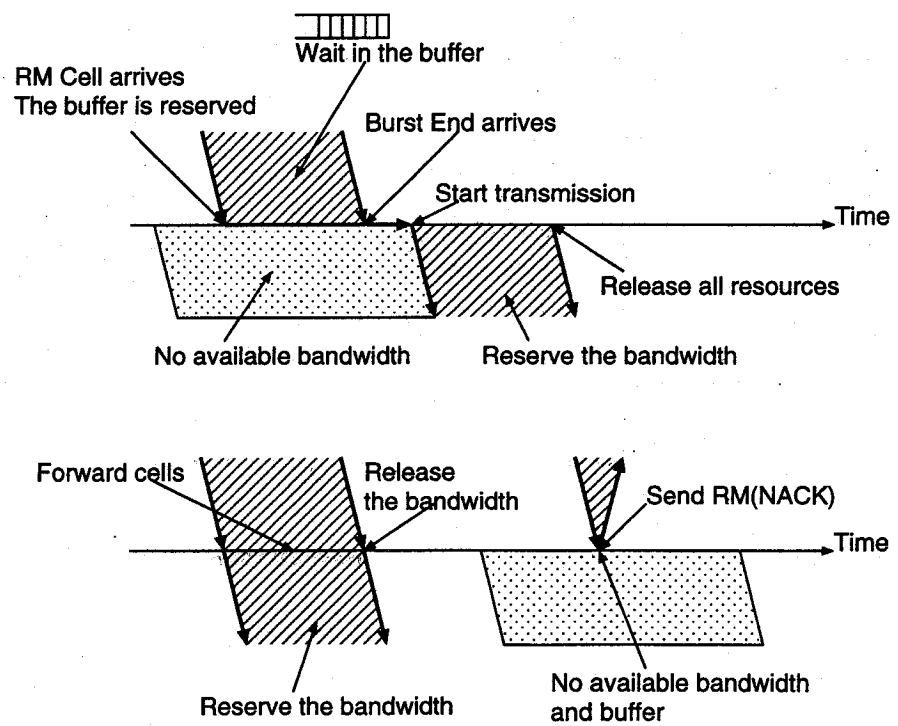


Figure 3.2: Timing Chart of Buffered ABT/IT

In our protocol, the switch does not reject the burst, instead it checks the available amount of buffer (F_a). If enough buffer is available for storing the entire burst (i.e., $F_a \geq l$), the switch reserves the buffer for the burst by setting $F_a \leftarrow F_a - l$. Then it stores the burst following the RM cell (see the upper part of Fig. 3.2).

If the buffer is too small to store the burst (i.e., $F_a < l$), the switch discards the burst and returns the backward RM(NACK) cell to the switch in the upstream (see the lower part of Fig. 3.2). Note that the backward RM(NACK) cell is not sent from the destination, but from the switch. Note that while this mechanism is not supported in the original ABT/IT (See also Fig. 1.1(b)), it is necessary to reduce the holding time of bandwidth/buffer reservations in the case of reservation failures. When the switch receives the backward RM(NACK) from the downstream, the switch releases the reserved bandwidth and buffer by setting $B_a \leftarrow B_a + b$ and $F_a \leftarrow F_a + l$, respectively.

Figure 3.3 summarizes the switch algorithm of buffered–ABT/IT.

3.2 Approximate Analysis

In this section, we present an approximate analysis for buffered ABT/IT protocol for networks of general topology. The analytical model is similar to the one for the virtual cut–through packet switching network [53, 54], but we extend the analysis to provide more accurate results.

3.2.1 Model

Consider a network with J links, labeled 1, 2, ... J. Link j ($j = 1, 2, \ldots J$) has bandwidth B. We assume that P asymmetric one-way connections of source-destination pairs have been already established in the network, and we do not consider new connection setups/releases. Connections are labeled 1 to P. Note that no bandwidth is reserved when the connection is established. We also assume that all bursts

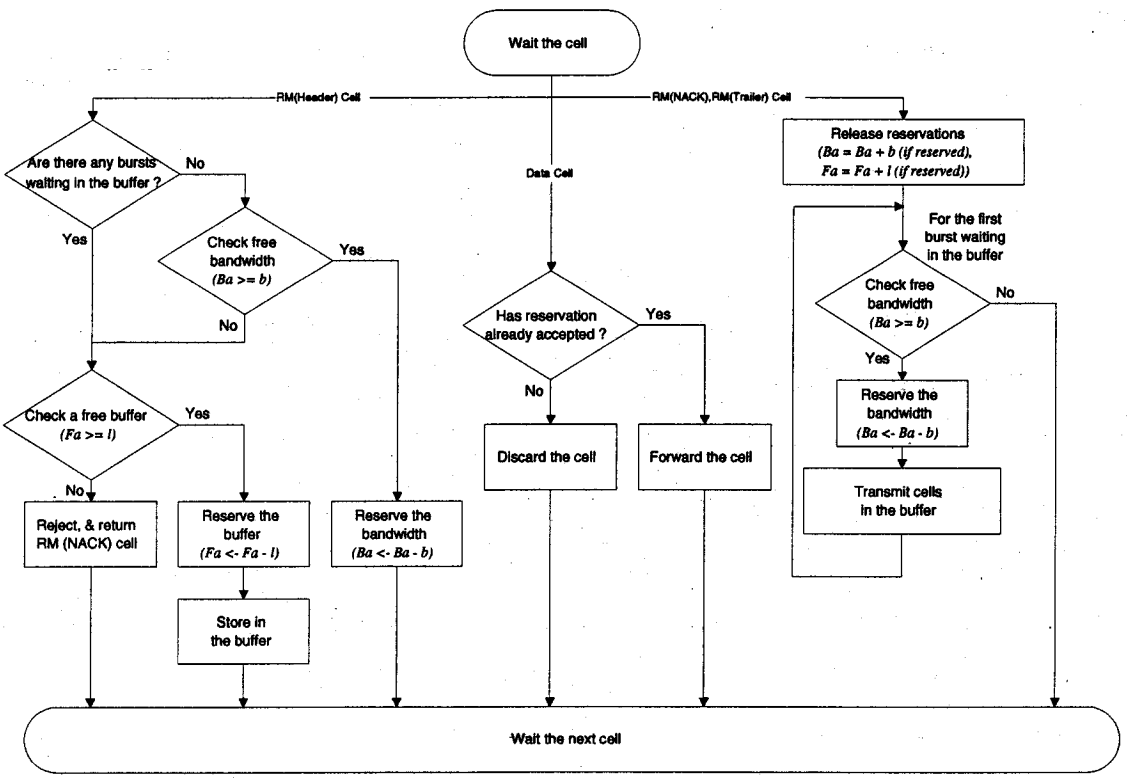


Figure 3.3: Switch Algorithm of Buffered–ABT/IT

are transmitted at b Mbps, where we assume $Nb = B$ for some integer N.

To analyze the performance of the above network, we assume the following. All links are assumed to have the same length and we denote the one–way propagation delay of a link by D. Let $J^{(p)}$ $(p = 1, \ldots, P)$ denote the set of links on the route of connection p. For $j \in J^{(p)}$, let $J_{j+}^{(p)}$ (resp. $J_{j-}^{(p)}$) denote the set of links between the source (resp. the destination) and link j on the route. Thus, $J^{(p)} = J_{j+}^{(p)} \cup j \cup J_{j-}^{(p)}$ for $j \in J^{(p)}$. Let $l_{p,1}$ denote the first link of connection p and R_j $(j = 1, \ldots, J)$ denote the set of connections having link j on their routes. Bursts on the pth connection are generated according to a Poisson process with rate λ_p. Lengths of bursts of all connections are independent and identically distributed according to the exponential distribution with mean $1/\mu$ (Mbit). Thus transmission times of all bursts are exponentially distributed with mean (μb) (sec). We assume that overheads of RM cell transmissions are negligible. We also assume that the network is stable and has reached its steady state in the rest of the paper. Furthermore, we assume that RM cell transmission have a priority over any other cell transmissions to transmit all RM cells without any losses.

The mathematical model is similar to that of a classical virtual cut–through packet switching network [53, 54], but we extend the analysis as follows. We assume the buffer size is finite, and the switch can serve multiple bursts simultaneously. Then, each switch is modeled by an $M/M/N/N + m$ queue. In our context, however, we implicitly assume that the buffer size m is counted in bursts, not cells. Thus one can utilize the analysis presented below by setting $m = N\mu$, where N denotes the buffer size in cells. In Section 3.3, we will validate the accuracy of this approximation on the buffer size through numerical experiments.

3.2.2 Blocking Probabilities and Transfer Delays Analysis

This subsection provides analytical formulas for the blocking probability and the transfer delay. Here, the transfer delay is defined as a time duration from a burst

generation at the source to its successful reception at the destination, given that the burst is successfully transferred to the destination.

As stated above, we approximately model the behavior of each link by an independent $M/M/N/N+m$ queue, where the traffic intensity at each link is reduced by considering blockings in upstream nodes. Let E_j denote the blocking probability of link j. We define $\lambda_{p,j}$ as the effective arrival rate of connection p in link j:

$$\lambda_{p,j} = \lambda_p \prod_{i \in J_{j+}^{(p)}} (1 - E_i). \tag{3.1}$$

The traffic intensity ρ_j of the aggregation of streams in link j is then given by

$$\rho_j = \frac{\sum_{i \in R_j} \lambda_{i,j}}{b\mu}. \tag{3.2}$$

Thus the steady state probability $p_j(k)$ of k bursts in the jth ($i = 1, 2 \ldots J$) node is approximately obtained as

$$p_j(k) = \begin{cases} \dfrac{\rho_j^k}{k!} p_j(0), & k = 0, \ldots, N-1, \\ \dfrac{\rho_j^k}{N! N^{k-N}} p_j(0), & k = N, N+1, \ldots, N+m, \\ 0, & k > N+m, \end{cases} \tag{3.3}$$

where

$$p_j(0) = \left[\sum_{k=0}^{N-1} \frac{\rho_j^k}{k!} + \frac{\rho_j^N}{(N-1)!} \frac{1 - (\frac{\rho_j}{N})^{m+1}}{(N - \rho_j)} \right]^{-1}. \tag{3.4}$$

The blocking probability E_j of link j $(1, \ldots, J)$ is then given by

$$E_j = \Pr[N(t) = N+m] = p_j(N+m). \tag{3.5}$$

We now provide an iterative procedure to obtain the blocking probability E_j and intensity ρ_j with equations (3.1) to (3.5). In what follows, for any symbol X, we denote the value of X in the nth iteration by ${}_{(n)}X$.

Step 1. Initial input: for all $p = 1, \ldots, P$ and all $j = 1, \ldots, J$,

i) Let ${}_{(0)}E_j = 0$

ii) Set a nonnegative small value to ϵ (e.g., $\epsilon = 10^{-3}$ for graphical representations).

iii) Let $n = 1$.

Step 2. The nth iteration:

i) Compute intensity ρ_j (3.2) for all $j = 1, \ldots, J$ with ${}_{(n-1)}E_j$.

ii) Compute the steady state probability $p_j(k)$ of k bursts in the jth ($i = 1, 2 \ldots J$) node with (3.3) and let ${}_{(n)}E_j$ be the resulting value.

Step 3. Convergence check

i) Let

$$Z = \sum_{p=1}^{P} |{}_{(n)}E_j - {}_{(n-1)}E_j| / {}_{(n)}E_j.$$

ii) If $Z \leq \epsilon$, we adopt ${}_{(n)}E_j$ ($p = 1, \ldots, P$) as approximate solutions to θ_p. Otherwise, add one to n and go to Step 2.

Even though we could not prove the convergence of the above iterative procedure, it converged in all of our numerical experiments shown in Section 4.

Let N_j be the mean number of bursts waiting in the buffer for link j ($j = 1, \ldots J$):

$$N_j = \sum_{k=N+1}^{N+m} (k - N) p_j(k).$$

Thus, with Little's Law, the mean waiting time L_j in the buffer for link j is given by

$$L_j = \frac{N_j}{(1 - E_j)(\sum_{i \in R_j} \lambda_{i,j})}.$$

The mean end-to-end delay can be obtained as follows. Let $Q_j(k)$ be the probability that the total number of bursts served in upstream links connected with link j is equal to k. To obtain $Q_j(k)$, we define the following variables. We define S_j as a set of upstream links connected with j. Let t_j be the number of such links: $t_j = |S_j|$.

Hereafter, links in $\boldsymbol{S}_j$ are labeled $s_{j,k}$ $(k = 1, \ldots, t_j)$. Furthermore, we define a $1 \times t_j$ vector $\boldsymbol{a}_j$ as

$$\boldsymbol{a}_j = (a_{j,1}, a_{j,2}, \ldots a_{j,t_j}),$$

where $a_{j,k}$ denotes the number of bursts served in upstream link $s_{j,k} \in \boldsymbol{S}_j$. Let $|\boldsymbol{a}_j|$ be

$$|\boldsymbol{a}_j| = \sum_{i=1}^{t_j} a_{j,i}.$$

Thus, assuming that each link is independent of others, we obtain the probability $Q_j(k)$ to be

$$Q_j(k) = \sum_{|\boldsymbol{a}_j|=k} \prod_{i=1}^{t_j} p_{s_{j,i}}(a_{j,i}).$$

We now define c_j as the probability that a randomly chosen burst going via link j cannot cut through link j (i.e., being forced to wait in the buffer before the bandwidth reservation). We assume that all connections $p(p \in R_j)$ have the same probability of c_j. Under the independence assumption of each link, c_j is given by $1 - \sum_{i=0}^{N-1} p_j(i)$. However our numerical experiments show that the results with this c_j are not accurate. To improve the accuracy of the approximation, we introduce the dependency between adjacent links.

We define α_j as the probability that a randomly chosen burst transmitted in upstream link i $(i \in \boldsymbol{S}_j)$ of link j cuts through link j. In what follows, by considering the dependency between adjacent links, we derive two equations for c_j and α_j to obtain the mean end-to-end transfer delay. Let R_j^+ denote a set of connections having link j on their routes and link j is not the first link on their routes. Then $\sum_{p \in R_j^+} \lambda_{p,j}$ denotes the arrival rate of bursts to link j, whose routes include an upstream link of link j. On the other hand, the total arrival rate to upstream links of link j is given by $\sum_{k \in \boldsymbol{S}_j} \sum_{p \in R_k} (1 - E_k)\lambda_{p,k}$. Note that this rate includes all connections coming from an arbitrary upstream link of link j either the connection goes to link j or not. Thus

α_j is obtained to be

$$\alpha_j = \frac{(1-c_j)\sum_{p\in R_j^+}\lambda_{p,j}}{\sum_{k\in \boldsymbol{S}_j}\sum_{p\in R_k}(1-E_k)\lambda_{p,k}}. \tag{3.6}$$

Let $q(l \mid k)$ denote the conditional probability that there are l burst cutting through link j given that k bursts are transmitted in upstream links of link j. We then have

$$q(l \mid k) = \binom{k}{l}\alpha_j^l(1-\alpha_j)^{k-l}, \tag{3.7}$$

and the conditional probability $r(l \mid k)$ that there are l burst cutting through link j given that k bursts are transmitted in link j is given by

$$r(l \mid k) = \binom{k}{l}(1-c_j)^l c_j^{k-l}. \tag{3.8}$$

Note here that given l burst are cutting through link j, the conditional probability $C(l)$ that a burst cannot cut through link j is given by

$$C(l) = \left(1-\sum_{i=0}^{N-1}p_j(i)\right)r(l|N) \Big/ \left[\sum_{i=l}^{N-1}p_j(i)r(l|i) + \left(1-\sum_{i=0}^{N-1}p_j(i)\right)r(l|N)\right]. \tag{3.9}$$

Therefore, by conditioning the total number of bursts transmitted in upstream links of link j, we have

$$c_j = \sum_{k=0}^{N-2}Q_j(k)\sum_{l=0}^{k}q(l \mid k)C(l) + \left(1-\sum_{k=0}^{N-2}Q_j(k)\right)\sum_{l=0}^{N-1}q(l \mid N-1)C(l). \tag{3.10}$$

We can obtain c_j, $q(l \mid k)$ and $q(l \mid k)$ with equations (3.6) -(3.10) by using iterative numerical procedure.

We are in a position to derive the mean transfer delay W_p of connection p. Suppose that a burst of connection p is not lost and eventually transmitted to the destination node. If this burst cut through all links on the path without waiting at intermediate nodes, the mean transfer delay is given by the sum of (i) the mean waiting time $L_{l_{p,1}}$ at the first node, (ii) the mean transmission delay $1/b\mu$ and the

total propagation delay $|J^{(p)}|D$, where $|J^{(p)}|$ denotes the number of links on the path of connection p. On the other hand, if a burst of connection p waits at the buffer of an intermediate node i, the additional delay (whose mean is given by L_i) incurs. Note here that c_j is considered as the joint probability of two events: (i) the burst is discarded at link j and (ii) the burst should wait in the buffer for link j. Hence, the probability of waiting in the buffer for link j is given by $c_j - E_j$. The mean transfer delay W_p of connection p is finally given by

$$W_p = L_{l_{p,1}} + |J^{(p)}|D + \frac{1}{b\mu} + \sum_{i \in J^{(p)}_{l_{p,1}-}} (c_i - E_i)L_i. \tag{3.11}$$

3.3 Numerical Discussions

The network models that we will use for numerical discussions are presented in Subsection 3.3.1. The accuracy of our approximate analysis is assessed in Subsection 3.3.2. Using a rather simple network model, we first discuss the effect of the buffer/bandwidth reservation in depth in Subsection 3.3.3. Observations made in Subsection 3.3.3 are then assessed using more general network topologies in Subsection 3.3.4. A flexible bandwidth reduction mechanism discussed in [32, 31] is finally applied to our buffered–ABT/IT protocol to investigate further performance improvement while results show that it is not necessary in our case mainly because the performance of buffered–ABT/IT itself is high enough.

3.3.1 Network Model

For the simple network model, we consider the tandem network model with two links shown in Fig. 3.4, which will be used Subsection 3.3.2 through 3.3.3 Subsection 3.3.1 and Subsection 3.3.4. As shown in Fig. 3.4, the two–hop connection C1 contends with the one–hop connection C2 for link L1, and with another one–hop connection C3 for link L2. Two ATM links, B_j (j = L1, L2), have the same bandwidth, 150 Mbps. The generation rate of bursts at sources are identically set to be

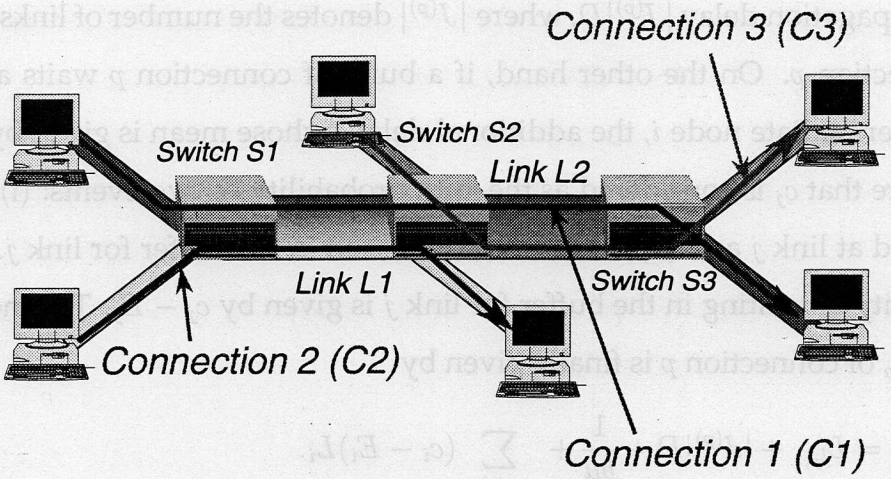


Figure 3.4: Network Model

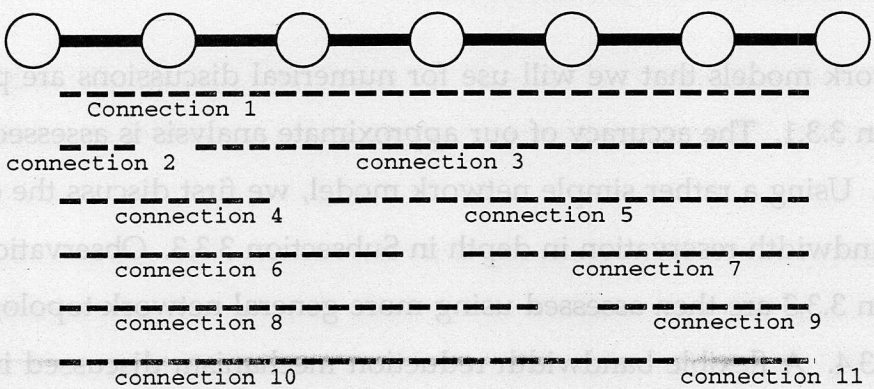


Figure 3.5: 6 Links Tandem Network Topology

λ_0, i.e., $\lambda_p = \lambda_0$ (p = C1, C2, C3). The mean burst length, $1/\mu$, is 5 Kbits, corresponding to 33 μsec on 150 Mbps link. The propagation delays of links L1 and L2 will be varied from 1 μsec to 1 msec.

A tandem network model with six links is used in Subsection 3.3.4 (Fig. 3.5). to examine the performance of long–hop connection because its performance in the original ABT/IT is much degraded as described in Section 1.2.

A more general network topology is also used in Subsections 3.3.2 and 3.3.4. We use an MCI–OC3 network topology [55] shown in Fig. 3.6. The network has 9 nodes

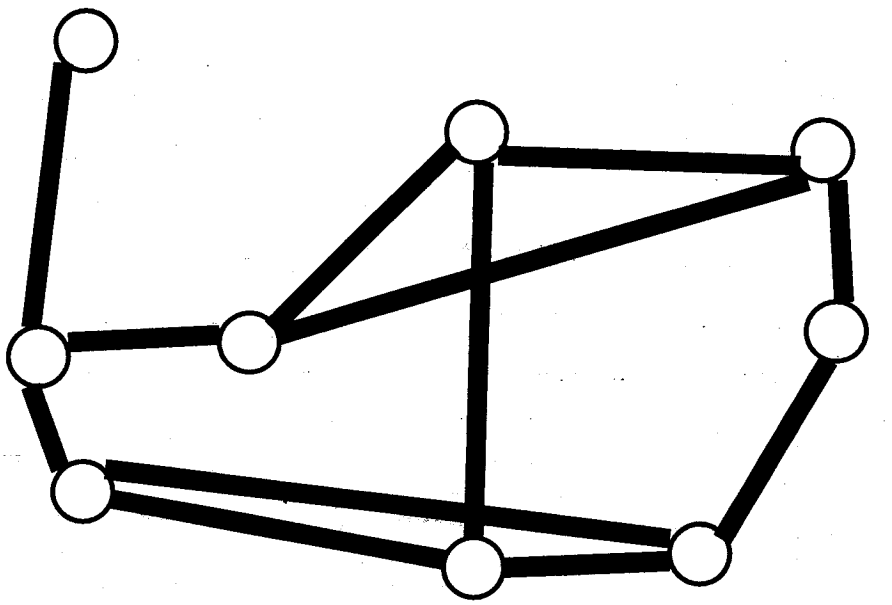


Figure 3.6: MCI–OC3 Network Topology

and 12 links. In the model setting, we assume that a single terminal is connected to each node. Then we set the one-way connection between every two terminals, which results in that the network has 36 connections in total. The connection is established on the shortest–path. If there are multiple shortest–paths with identical length, the route is chosen randomly.

3.3.2 Accuracies of Analysis

We first assess the accuracies of our approximate analysis by comparing with simulation. In Figs. 3.7 and 3.8, we compare the throughput and the transmission delay of each connection. Here, we set the propagation delay of each link (Links L1 and L2) to be 1 msec. The requesting bandwidth b is set to be 75 Mbps. In our simulation experiments, 200,000 bursts are generated from each terminal. We can observe a good agreement between analysis and simulation results.

We next validate the accuracy of the assumption that the buffer size is counted in bursts, not cells. Given the buffer size in bursts, F, and the mean burst length,

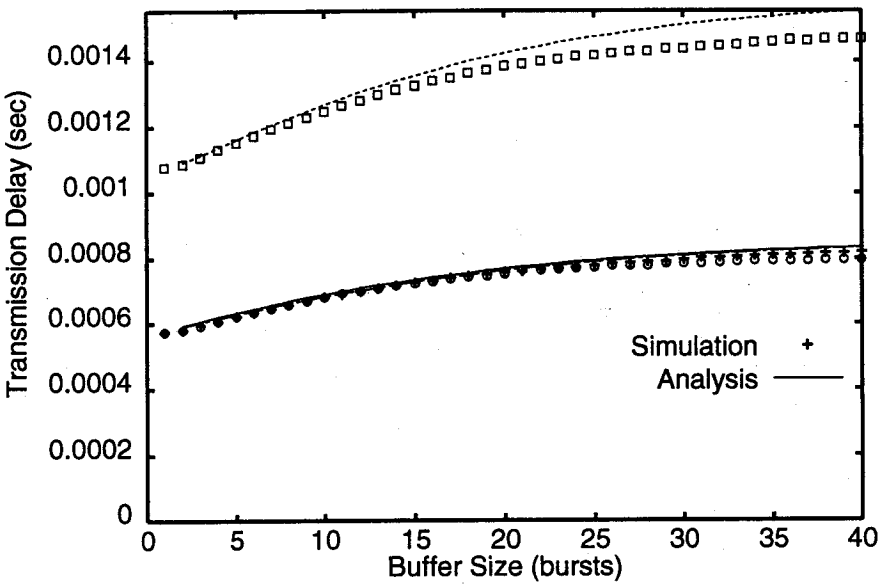


Figure 3.7: Accuracy of Transmission Delays

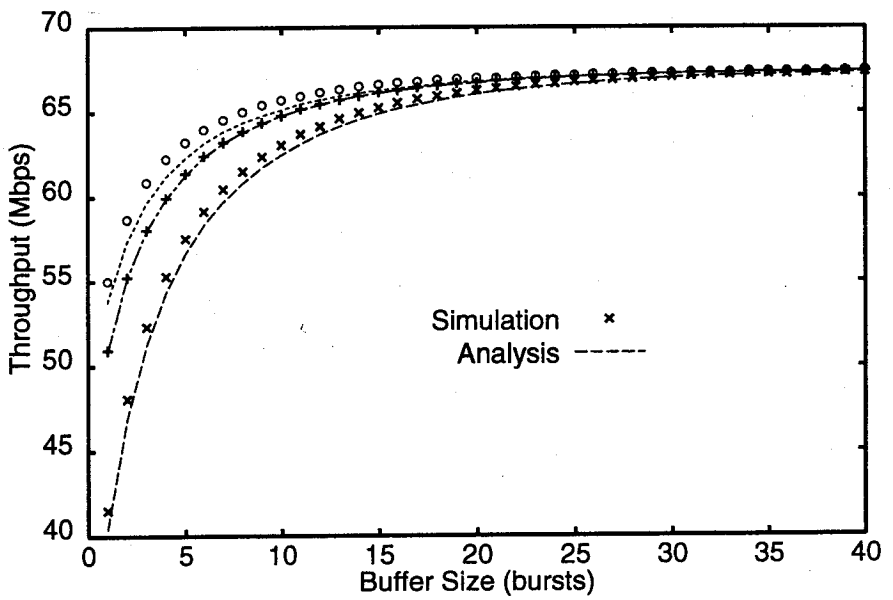


Figure 3.8: Accuracy of Throughput

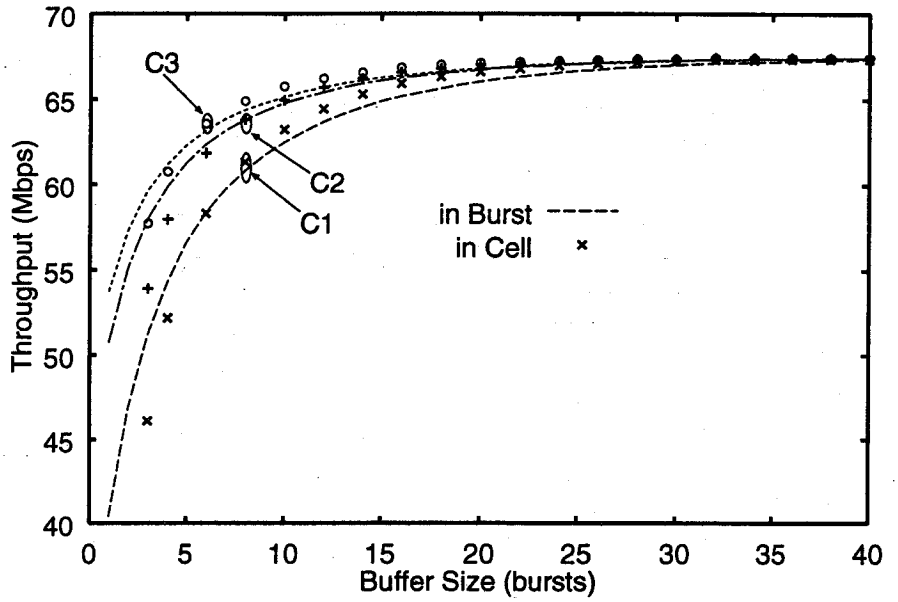


Figure 3.9: Comparison Between Buffer Units; in Bursts vs. in Cells

$1/\mu$, we set the buffer size to be $F/(424\mu)$ and the average number of cells in the burst $1/(424\mu)$ in the simulation. In our results, the mean burst length is set to be 5 Kbits, corresponding to 13 ATM cells, and the mean number of bursts which can be stored in the buffer is $F/5$K (bursts). In Fig. 3.9, we compare the results. Lines in the figure indicate the analysis results in the case where the unit is bursts, and dots show the simulation results in which the unit is cell accordingly to our actual protocol proposal. As shown in this figure, the accuracy is good except in the case of very small buffer size.

A more general network topology described in Subsection 3.3.1 was also examined. Comparisons of the throughput between analysis and simulation are shown in Fig. 3.10 where the horizontal axis shows the connection numbers 1 through 36. The mean transmission delays are also compared in Fig. 3.11. We can again observe good accuracies in the figures.

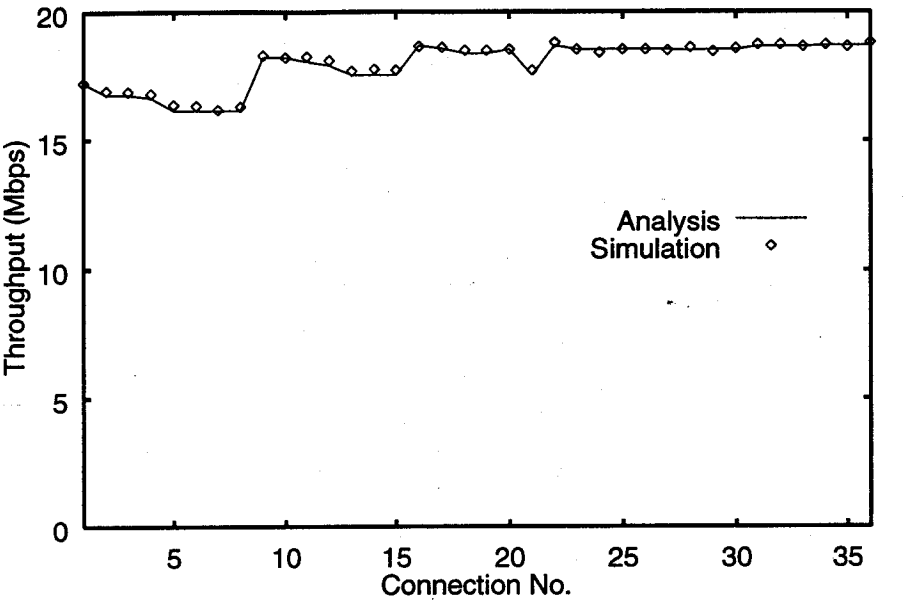


Figure 3.10: Accuracy of Throughput in MCI-OC3 Network

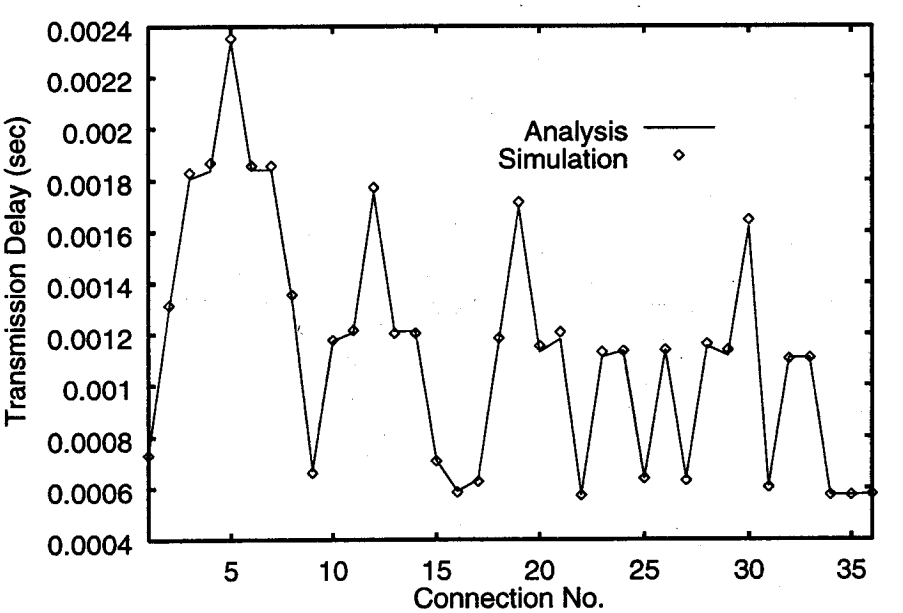


Figure 3.11: Accuracy of Transmission Delays in MCI-OC3 Network

3.3.3 Effects of Buffered ABT/IT

We first examine how much the throughput of ABT/IT can be improved by buffer reservation using the tandem network model depicted in Fig. 3.4. Figure 3.12 compares the throughput of ABT/IT with and without buffer reservation. The horizontal axis is the offered load which is identically set among connections. The requesting bandwidth is identically set to be 75 Mbps for every source, and the propagation delay be 1 msec. The buffer size is 10 in bursts. Furthermore, the result of the original ABT/IT is obtained from [32, 31]. As shown in the figure, the throughput of ABT/IT with buffer reservation can be much improved so that it becomes hard to observe differences among connections C1 through C3 except for very high traffic load condition. It is noteworthy that the throughput of two-hop connection C1 in buffered–ABT/IT becomes two times larger than that in the original ABT/IT.

We next show how much buffer is needed to obtain a high throughput. Figure 3.13 shows the obtained throughput of each connection depending on the buffer size. The offered load of each connection is set to be high, i.e., 74 Mbps (approximately 99% traffic load on each link), and the propagation delay is set to be long, i.e., 1 msec. We note that while the required buffer size must depend on the traffic load and/or the propagation delays, the parameter setting in this example shows a rather extreme case. From the figure, we can observe that the throughput can exceed 70 Mbps when the buffer size is beyond 10 or 20. For example, if the mean burst length is 100 Kbits (corresponding to 250 ATM cells), the necessary buffer size becomes 2,500 or 5,000 cells, which is easy to implement even with the current switch technology. While we do not show results due to space limitation, we have confirmed that the observation on the required buffer size is independent of the burst length.

We last note that improved throughput leads to larger transmission delays by storing bursts at the buffer of intermediate switches. Figure 3.14 compares the transmission delay against the offered load for ABT/IT with and without buffer reserva-

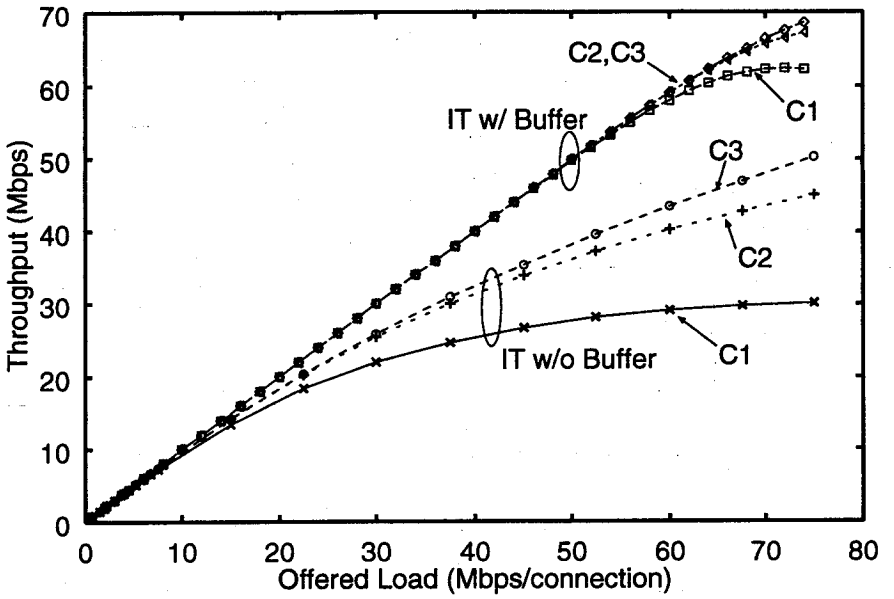


Figure 3.12: Throughput Comparisons with and without Buffer (Propagation Delay = 1 msec)

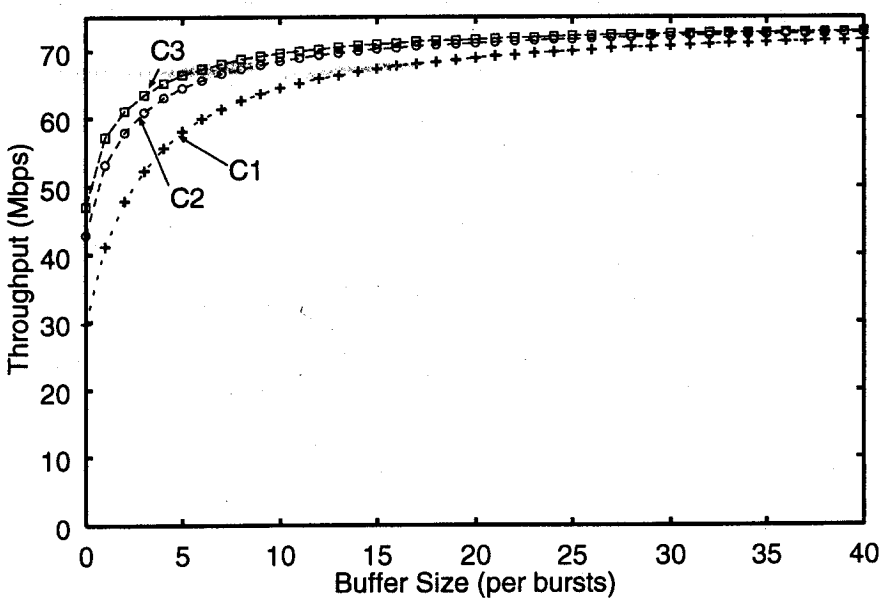


Figure 3.13: Effect of Buffer Size on Throughput

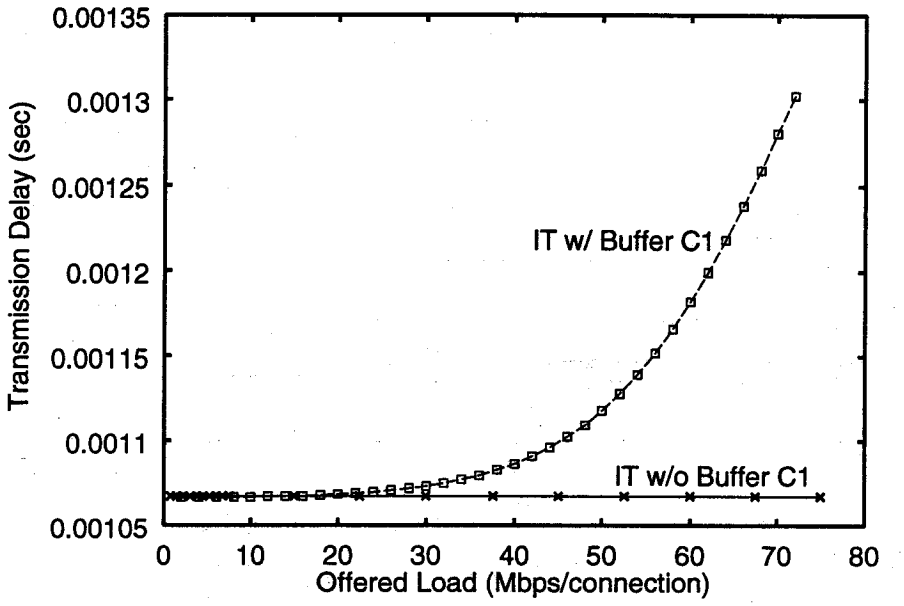


Figure 3.14: Transmission Delay Comparisons (Connection C1)

tion. To obtain this figure, we set the buffer size to 10 in bursts, and the propagation delay is 1 msec. The influence of the buffer size is clearly shown in Fig. 3.15 where the traffic load is set to be 74 Mbps.

3.3.4 Cases of General Topologies

In this subsection, we examine more general network topologies. For this purpose, we first use the six–link tandem network which has 11 connections (Fig. 3.5). The propagation delay is set to be 1 msec for each link. The mean throughput as a function of the number of hops is plotted in Fig. 3.16 where the offered load is set to be 25 Mbps/Connection. As shown in the figure, we can observe significant throughput improvement even for long–hop connections. For the connection having six links (Connection 1), the buffered–ABT/IT can offer five times larger throughput than the original ABT/IT.

We next apply our protocol to the MCI–OC3 network topology shown in Fig. 3.6. The connections are established as described in Subsection 3.3.1. That is, the con-

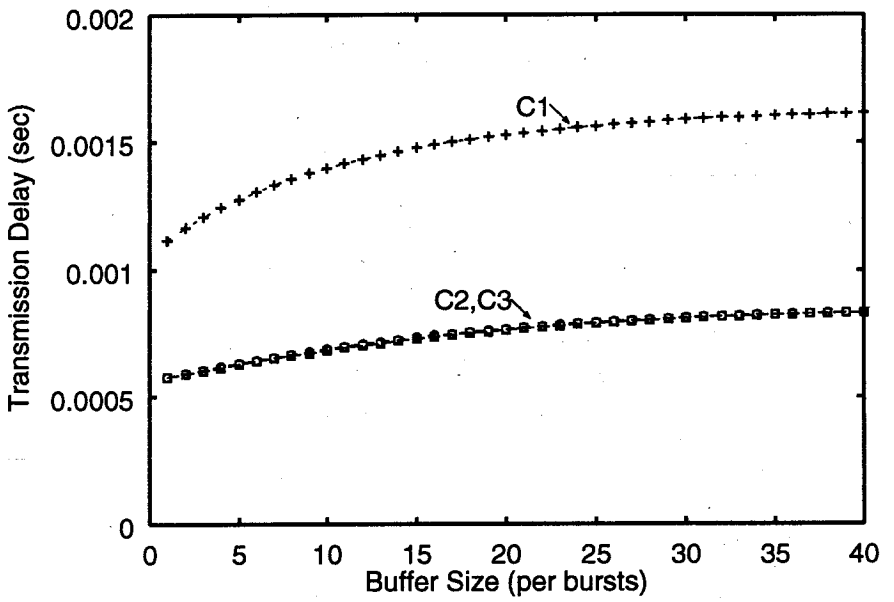


Figure 3.15: Effect of Buffer Size on Transmission Delay

nection between every two nodes is established with the shortest–path, and the route of the connection is randomly chosen if there are multiple shortest–paths with identical hop count. Figure 3.17 shows the throughput for each connection. The propagation delay is set to be 1 msec for each link and the offered load is set to be 18.7 Mbps/Connection. Connection 1 traverses four hopss, and connections 2 through 7 traverses three hops. From this figure, we can also observe throughput improvement even for long–hop connections. Figure 3.17 also shows that the fairness of throughput between connections becomes improved. In the original ABT/IT case, throughput of each connection changes drastically even between connections with the same number of hops, while the difference among connections becomes small in buffered–ABT/IT.

3.3.5 Effect of Flexible Bandwidth Reservation Mechanism

In this subsection, we investigate the effect of the bandwidth reduction mechanism introduced in [32, 31]. The bandwidth reduction mechanism is performed as fol-

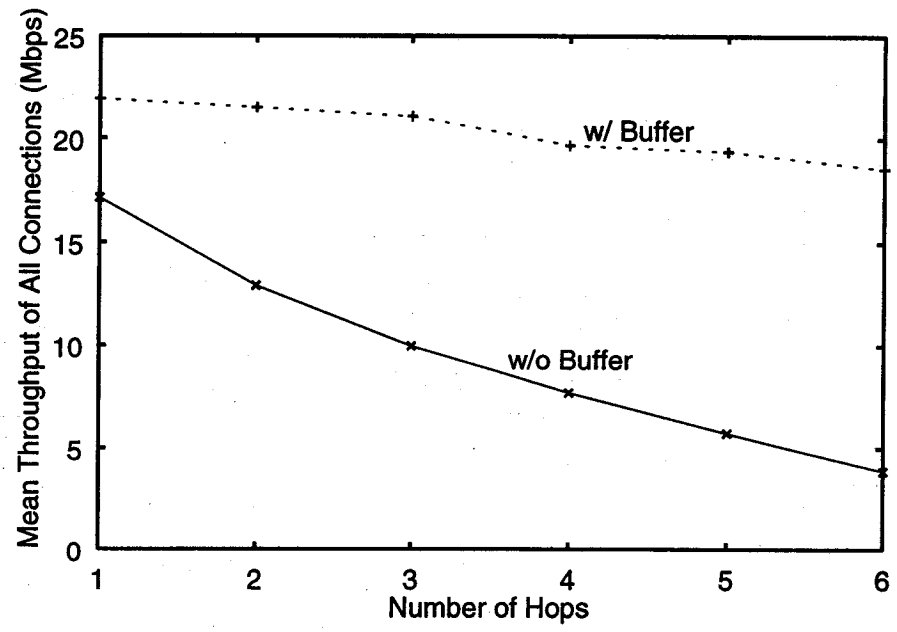


Figure 3.16: Average Throughput Dependent on the Number of Hops

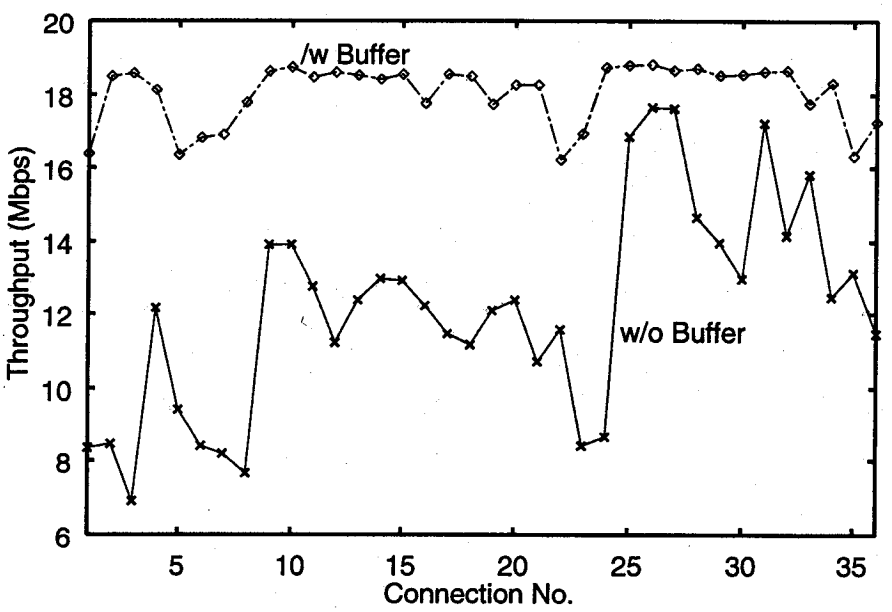


Figure 3.17: Throughput of Connections – MCI Network Topology

lows. Each source requests the bandwidth with an initial value, b, by using the forward RM cell. If the requested bandwidth b is available on the link, the switch simply accepts the request. On the other hand, if the available bandwidth of the link is smaller than that value, the switch checks whether half of the requested bandwidth is available or not. If it is available, the switch reserves the bandwidth of $b/2$ and overwrites it in the forward RM cell. If not, the switch again checks whether another half ($b/4$) is available or not. In this way, the bandwidth is reduced until the available bandwidth is found. Such a mechanism has been studied in [32, 31] and it was shown that the performance can be improved to some extent in the case of ABT/IT without buffer reservation. With this mechanism, the blocking probability of the buffered ABT/IT is expected to be further decreased.

In our buffered–ABT/IT, the following changes are necessary in order to apply the above bandwidth reduction mechanism. When the requesting bandwidth is reduced by the switch, it may still exceed the available bandwidth. At that time, a part of the burst, which cannot be sent with the reduced bandwidth, is stored in the buffer if the buffer is available. Namely, the buffer of the buffered–ABT/IT is utilized (1) if the bandwidth is full or (2) if the bandwidth reserved by the bandwidth reduction mechanism is less than the requested bandwidth.

We conducted simulation experiments using the two–link tandem network model shown in Fig. 3.4. The results showed that the effect is quite limited in the case of buffered–ABT/IT since buffering of bursts has already offered the improved throughput. This can be seen from Fig. 3.18, showing the blocking probabilities of Links L1 and L2 in the buffered–ABT/IT with and without bandwidth reduction. In obtaining the figure, the buffer size is set to be 0.1 Mbits. For comparison purposes results of the original ABT/IT with and without bandwidth reduction are also shown in the same figure. The bandwidth reduction mechanism can reduce the blocking probabilities in both cases of the original and buffered ABT/IT, but differences in the throughput in the buffered ABT/IT are quite small.

The buffer size does not affect this tendency as shown in Fig. 3.19 where the

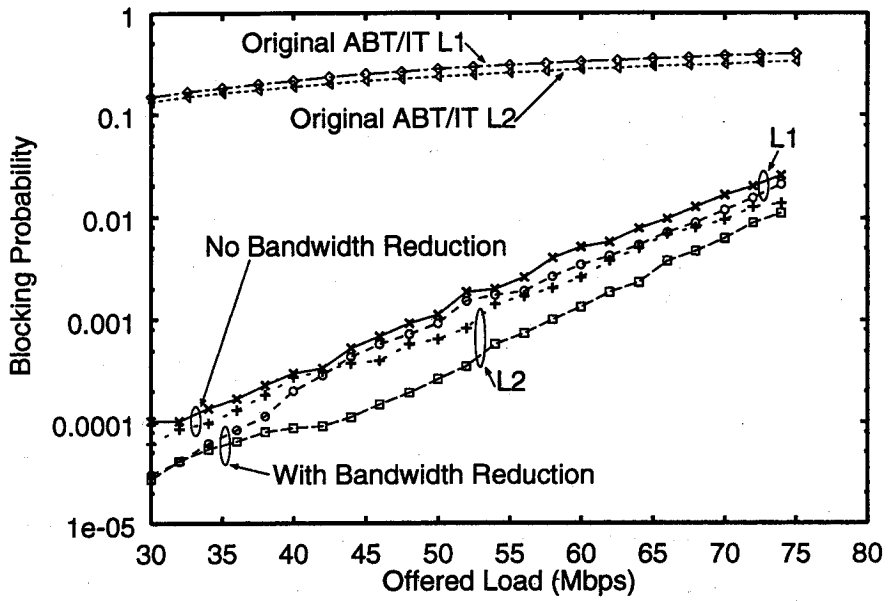


Figure 3.18: Effect of the Bandwidth Reduction; Blocking Probability

offered load is set to 74 Mbps. We can observe that larger buffer size leads to lower blocking probabilities, but the improvement is very limited. We note here that if the buffer size is small, the blocking probability of bandwidth reduction unexpectedly becomes worse. It is because the bandwidth reduction mechanism tends to share the bandwidth with more bursts and it leads to lower link utilization and higher buffer utilization in our case.

In summary, the bandwidth reduction mechanism can improve the performance in the original ABT/IT, but there is no significant improvements in the buffered–ABT/IT since the buffered–ABT/IT itself can provide a high throughput.

3.4 Concluding Remarks

In this paper, we have investigated the effect of buffer reservation in ABT/IT. For this purpose, we have proposed the buffered ABT/IT, and its approximate analysis is developed to obtain the throughput and the mean transfer delay. Then, we have shown that buffer reservation in ABT/IT can lead to significant performance

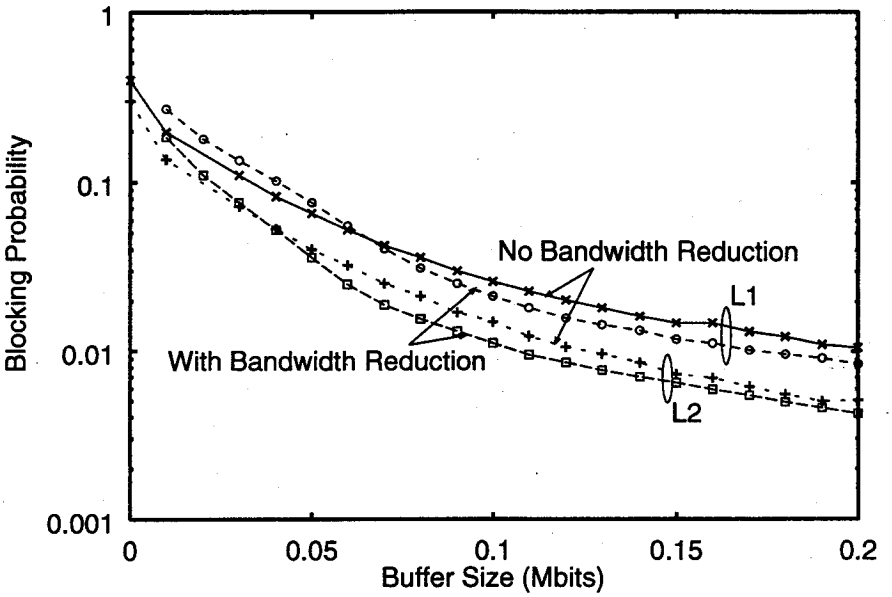


Figure 3.19: Effect of the Bandwidth Reduction in Buffer Size

improvement by comparing with the original ABT/IT protocol. We have also addressed the required buffer capacity for obtaining a high throughput. We have shown that it is a reasonable size based on the current switch technology. Finally, we have considered the bandwidth reduction mechanism which is helpful in improving the performance of the original ABT/IT protocol. However, in the case of the buffered–ABT/IT, it already provides a high throughput and the performance improvement with the bandwidth reduction mechanism is quite small. For future research topics, we may need to compare the buffered–ABT/IT with other protocols or service classes defined in ATM networks such as the ABR service class and the UBR service class.

Chapter 4

Performance Evaluation of TCP over ABT Protocols

Most of past studies focused on the data transfer capability of ABT within the ATM layer. In actual, however, we need to consider the upper layer transport protocol since the transport layer protocol also supports a network congestion control mechanism. One such example is TCP (Transmission Control Protocol), which is now widely used in the Internet. In this chapter, we evaluate the performance of TCP over ABT protocols. Simulation results show that the retransmission mechanism of ABT can effectively overlay the TCP congestion control mechanism so that TCP operates in a stable fashion and works well only as an error receovery mechanism.

4.1 Simulation Model

4.1.1 Reservation Mechanisms in ABT Protocols

In this subsection, we briefly introduce ABT/IT, ABT/DT, buffered ABT/IT and ABT/DT with bandwidth negotiation.

Bandwidth Reservation Mechanisms in ABT/IT and ABT/DT

In the original ABT protocol [1, 5, 6], the route of the connection is determined at the connection setup time, but the bandwidth is not reserved. When the burst actually arrives at the source, a forward RM (Resource Management) cell is sent to the destination for the bandwidth reservation along the predetermined route. The RM cell contains the required bandwidth to transfer the burst, and each switch on the route reserves the bandwidth. It then forwards the RM(ACK) cell to the next switch in the downstream. If a sufficient amount of the bandwidth is not available on the link, on the other hand, the switch forwards the RM(NACK) cell to the destination. The destination having received the forward RM(ACK) or RM(NACK) cell returns the RM(ACK) or RM(NACK) cell to the source as the backward RM cell. Every switch receiving the backward RM(NACK) cell releases the reserved bandwidth if it has reserved the bandwidth. The source receiving the backward RM cell can finally recognize whether the bandwidth request is admitted or not. This is called ABT with Delayed Transmission (ABT/DT), which is illustrated in Fig. 1.1(a).

When the link bandwidth becomes large as in recent high speed networks, the overhead time to wait the backward RM cell before the burst transmission is not acceptable. That is the reason why ABT/IT (ABT with Immediate Transmission) was introduced [5]. In ABT/IT, the source sends the burst immediately following the forward RM cell without waiting acknowledge of the reservation as shown in Fig. 1.1(b). If the sufficient bandwidth is available, each switch accepts the burst to forward it to the next switch in the downstream. If not, on the other hand, the switch rejects the incoming burst, and returns the backward RM(NACK) cell via the destination to notify the source that the burst is lost. While this mechanism introduces a hardware complexity to selectively discard the burst, it must alleviate the influence of the large propagation delay.

Bandwidth/Buffer Reservation Mechanism in Buffered ABT/IT

We next describe the bandwidth/buffer reservation mechanism in buffered ABT/IT [35]. When the burst arrives at the psource, the forward RM cell with the requesting bandwidth and the length of the burst is sent to the destination along the predefined route, followed by the burst transmission as in the original ABT/IT protocol. Each switch receiving the RM cell first checks the available bandwidth. If a sufficient amount of the bandwidth is available on the link, it reserves the bandwidth and forwards the RM cell to the next switch in the downstream. Thus, the operation is identical to the original ABT in this case. The different treatment is performed when the requesting bandwidth is not available. In buffered ABT/IT, the switch does not reject the burst, but checks the available amount of the buffer next. If the buffer is available for storing the entire burst, the switch reserves the buffer for the burst and stores the burst following the RM cell (see the upper part of Fig. 3.2).

If the buffer is also lack of storing the burst, the switch discards the burst and returns the backward RM(NACK) cell to the next switch as in the original ABT/IT (see the lower part of Fig. 3.2). When the switch receives the backward RM(NACK) from the downstream, the switch releases the reserved bandwidth and buffer.

ABT/DT with Bandwidth Negotiation

In ABT/DT, it is possible that the switch reserves the bandwidth less than the requested bandwidth specified in the RM cell if the latter is not available on the link. The switch then overwrites the new bandwidth on the RM cell to forward it to the next switch. In this bandwidth reservation method, each switch checks the bandwidth in the backward RM cell, and reduces the reserved bandwidth if the temporarily reserved bandwidth is larger than the one in the backward RM cell. The source receiving the backward RM(ACK) cell then starts to transmit the burst according to the bandwidth specified in the RM cell. A more detail of ABT/DT with bandwidth negotiation is as follows. Each source requests the bandwidth with

an initial value, b, by using the forward RM cell. If the requested bandwidth b is available on the link, the switch simply accepts the request. On the other hand, if the available bandwidth of the link is smaller than that value, the switch checks whether the half of the requested bandwidth is available or not. If it is available, the switch reserves the bandwidth of $b/2$ and overwrites it in the RM cell. If not, on the other hand, the switch again checks whether another half ($b/4$) is available or not. In this way, the bandwidth is reduced until the available bandwidth is found. When the switch receives the backward RM cell, it adjusts the reserved bandwidth to the one specified in the RM cell. Recall that such a negotiation cannot be implemented in ABT/IT since the burst is transmitted immediately following the RM cell. In evaluating the bandwidth negotiation mechanism presented in the next section, the initial requesting bandwidth b was set to be 150 Mbps. The minimum bandwidth was 150/16 Mbps. Namely, if the available bandwidth on the link is less than 150/16 Mbps, the reservation request is rejected. It prevents the reserved bandwidth from being much less than the requesting bandwidth.

4.1.2 Data Transport Mechanism in TCP over ABT

In this subsection, we describe the data transport mechanism of TCP over ABT (see Figure 4.1). It consists of three major parts; TCP layer, ABT service class layer, and ATM layer.

We consider that data transmission is performed as follows (see also Fig. 4.2). The path between source and destination is established in advance using the signaling protocol.

1. At the TCP sender, each TCP segment is passed to the ABT layer as long as the TCP window is allowed.

2. At the ABT layer, each segment from the TCP layer is treated as a single burst (see Fig. 4.3). The source ABT sends the RM cell according to the ABT protocol to transmit the burst. In the case of ABT/IT, the actual burst transmission

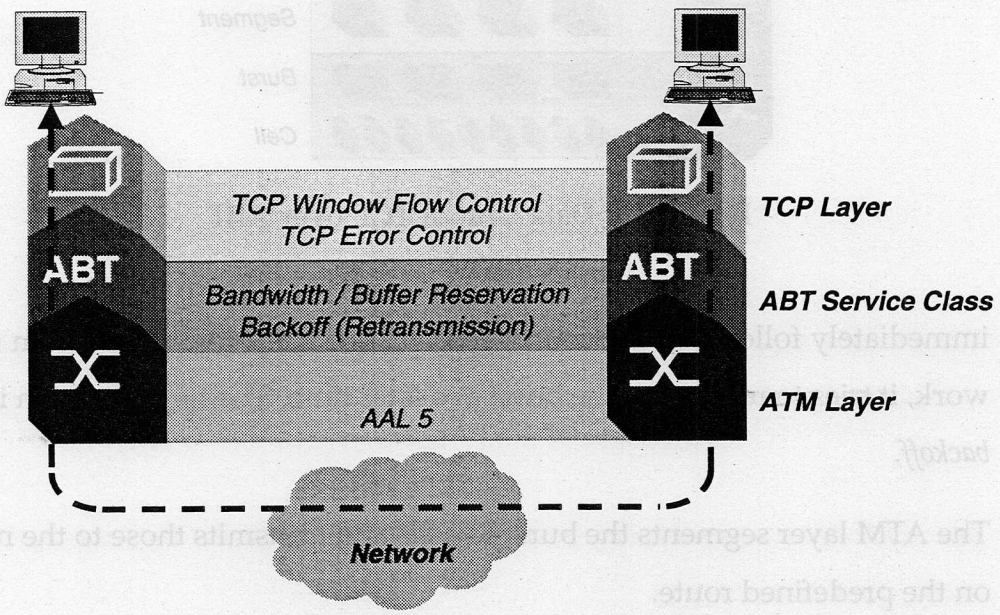


Figure 4.1: Network Architecture of TCP over ABT

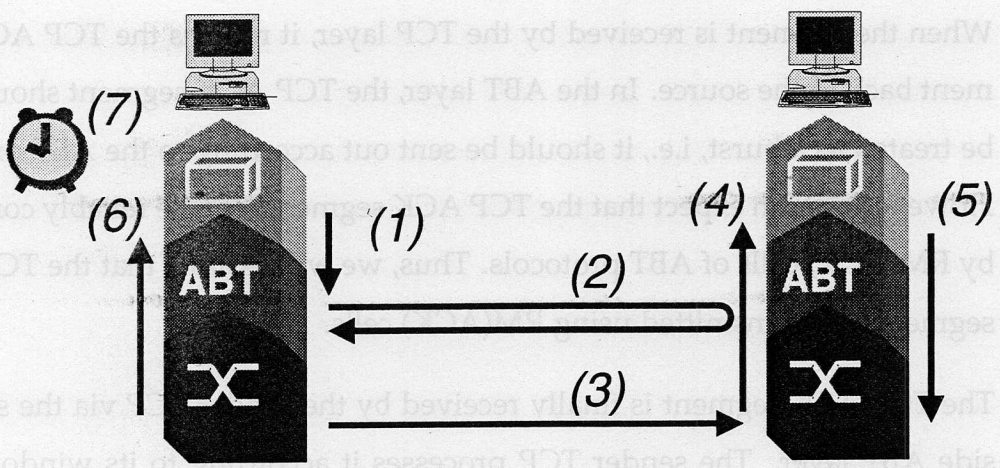


Figure 4.2: Transport Process

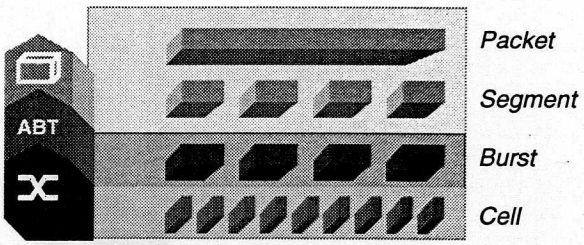


Figure 4.3: Data Unit of TCP over ABT

immediately follows. If the source receives the RM(NACK) cell from the network, it tries to retransmit the burst at a later time. The time duration is called *backoff*.

3. The ATM layer segments the burst to cells and transmits those to the network on the predefined route.

4. The ATM layer at destination assembles cells into the burst to pass it to the ABT layer.

5. The ABT layer simply forwards the burst to the TCP layer as the received segment.

6. When the segment is received by the TCP layer, it returns the TCP ACK segment back to the source. In the ABT layer, the TCP ACK segment should also be treated as a burst, i.e., it should be sent out according to the ABT protocol. However, we can expect that the TCP ACK segments can be reliably conveyed by RM(ACK) cells of ABT protocols. Thus, we will assume that the TCP ACK segments are transmitted using RM(ACK) cells.

7. The TCP ACK segment is finally received by the sender TCP via the sender–side ABT layer. The sender TCP processes it according to its window flow control mechanism.

In the case of TCP over EPD, the ABT layer does not exist at the end systems, but the special queuing scheduling is performed at the switching nodes [56].

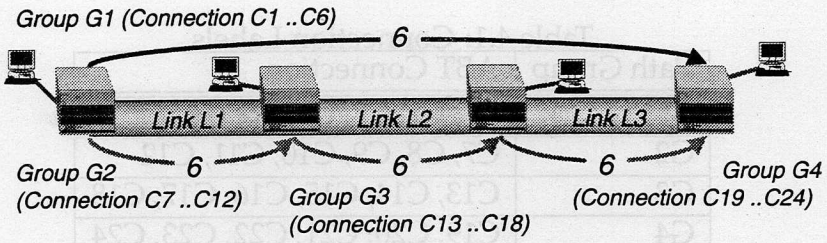


Figure 4.4: Three Tandem Network Model

Last, we note that as described in the above, we assume that the TCP segment corresponds to the burst in the ABT protocol. Since the segment is rather short, our assumption gives a performance penalty on ABT protocols because ABT was originally proposed for transferring rather long bursts, and it is convinced that ABT is valuable to such a case. However, we believe that the applicability to handle the short burst should also be validated for the ABT protocols to be widely used in the real world.

4.1.3 Network Model

In this subsection, we describe our network model for performance evaluation. We consider the tandem network model with three links shown in Fig. 4.4. The network has three ATM links in tandem and four ATM switching nodes. All ATM links have the same bandwidth; 150 Mbps. We set four groups (Group G1–G4) with different hop counts, each of which contains six ABT connections. That is, the network has 24 ABT connections in total and 12 ABT connections on each link. ABT connections are labeled as shown in Table 4.1. Each ABT connection supports one TCP connection. Every TCP connection is assumed to always have segments to transmit. By this assumption, we can investigate the maximum throughput.

The reference parameter values are summarized in Table 2. We will use these parameters unless otherwise stated explicitly. It is assumed that processing time at each layer is zero for simplicity.

Table 4.1: Connection Labels

Path Group	ABT Connection
G1	C1, C2, C3, C4, C5, C6
G2	C7, C8, C9, C10, C11, C12
G3	C13, C14, C15, C16, C17, C18
G4	C19, C20, C21, C22, C23, C24

Table 4.2: Default Values of Parameters

Parameter	Default Value
Propagation Delay	500 μsec/link
Buffer Size	800 Kbits (1887 ATM Cells)
Maximum Window Size	64 KBytes (512 KBits)
TCP Segment Size	4352 Bytes (34.816 KBits)
TCP Data Size	800 Kbits/TCP Conn. (Exponentially Distributed)
Requesting Bandwidth	37.5 Mbps/ABT Conn.

4.2 Simulation Results

In this section, we first investigate the performance of TCP over original ABT/IT and TCP over original ABT/DT. Unfortunately, those are not attractive since the simulation results show unacceptable degree of the unfairness among the connections with different hop counts. Further, those do not provide performance improvement when compared with the UBR with EPD case. One of main reasons is interactions of the backoff times in ABTs and the RTO retransmit timer implemented in TCP. Thus, we made some parameter tuning to obtain more throughput in TCP over ABT. Those results are presented in Subsection 4.2.1 for ABT/IT and in Subsection 4.2.2 for ABT/DT, respectively. Different from ABT/DT and IT, buffered ABT can give much performance improvement, which will be shown in Subsection 4.2.3.

4.2.1 Performance of TCP over ABT/IT

Our first result compares TCP over EPD and TCP over ABT/IT. In Fig. 4.5, the mean throughput of two cases are shown dependent on the propagation delays. In obtaining the figure, TCP retransmission timeout (RTO) is not changed. The mean backoff time of ABT/IT is identically set to be 928 μsec, which corresponds to the transmission time of one segment on 150 Mbps link. In the figure, the average throughput of connections in each group (G1 of three hop connections and G2, G3, G4 of one hop connections) is plotted. As shown in the figure, throughput of one hop connections is dramatically increased by using ABT/IT. Then, the total throughput of all connections is also increased dramatically. For example, the total throughput of TCP over ABT/IT is 90.18 Mbps while it is only 31.22 Mbps in TCP over EPD for 1 msec propagation delay. However, the figure also indicates the "unfairness" problem among connections with different hop counts in the ABT/IT case. The throughput of the connections in Group G1 (long hop connections) is almost zero in the ABT/IT case. It is because in ABT/IT, the short hop connections receives RM(NACK) cells faster than the long hop connections do. Then, the short hop connections can have a chance to retransmit the bursts more, which results in that long hop connections fail to transmit their bursts. Another reason, which is a fundamental problem in the TCP over ABT/IT case, is that once the burst is rejected at the switch (according to the ABT/IT protocol), the backoff is performed at the source. However, in the current parameter setting, TCP is also likely to be aware of loosing the segment, which results in closing the TCP window. Since the short hop connections can transmit the bursts successfully, decrease of the window size is likely to be performed only by the long hop connections.

It is well known that in the window flow control mechanism of the TCP layer, lost of segments drastically degrades the throughput performance since it is detected by the timeout mechanism. The reasons are as follows. First, the timeout threshold is rather long (four times of the variance of RTT [11]). Second, the window size

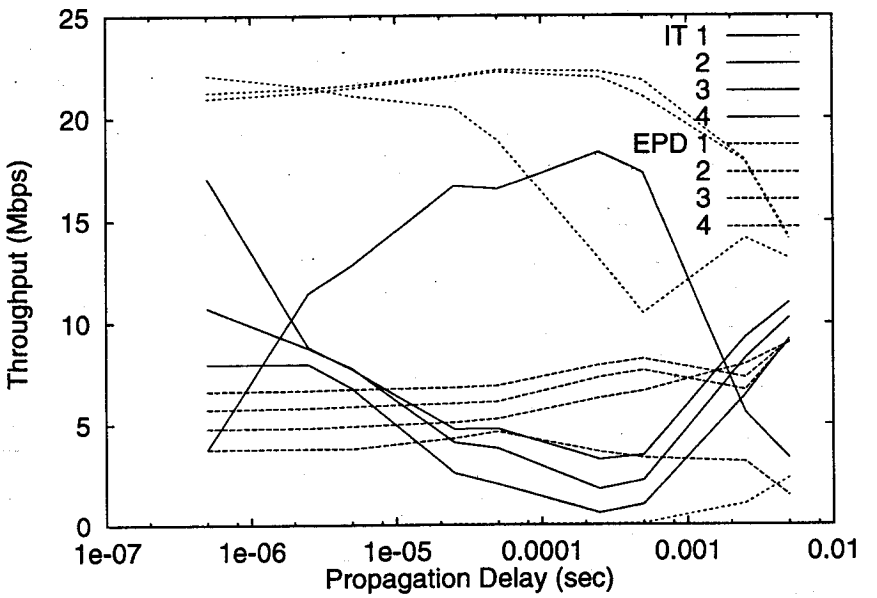


Figure 4.5: Comparisons between ABT/IT and EPD

is decreased to one in the TCP Tahoe version, and is set to be a half in the TCP Reno version if the first retransmit fails. Hence, it becomes appropriate that the ABT layer handles the segment retransmission (actually, the burst in the ABT layer). However, the TCP timeout mechanism is still necessary because the segment may be lost elsewhere. Thus, the backoff time should be carefully chosen such that lost bursts can be retransmitted as many as possible within the TCP timeout threshold. There are two ways to increase the number of backoffs; (1) to shorten the backoff time within the ABT layer, or (2) to lengthen the timeout threshold (RTO) of the TCP layer. The first solution is preferred since the TCP protocol has already widely used in the real network while the ABT protocol is not. However, we have found that the second solution can offer the slightly better performance, which will be shown next.

We first examine the impact of the backoff time. The backoff time affects how many times the source can backoff the segment within the timeout of the TCP layer. If the backoff time is too long, most of bursts cannot be retransmitted by the backoff, and it leads to performance degradation. If the backoff time is set to be short, on

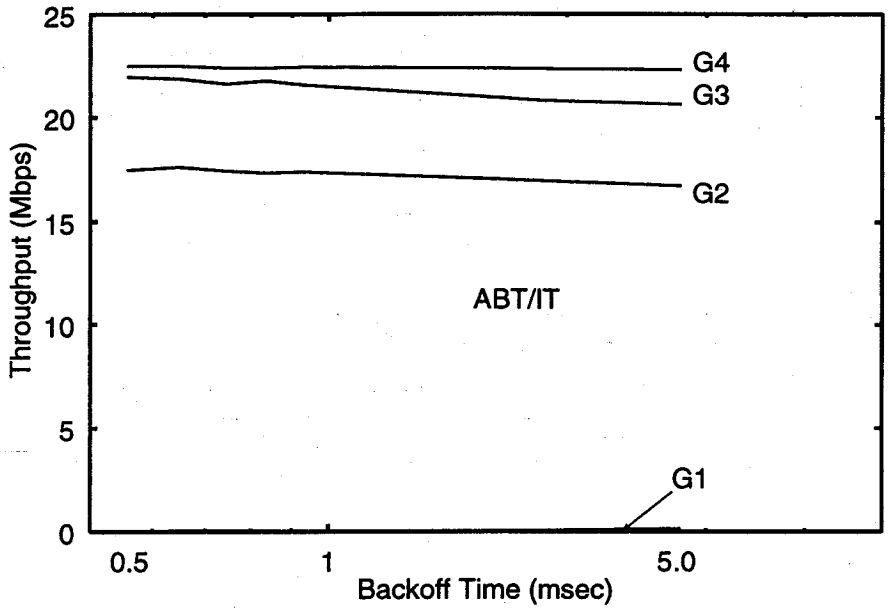


Figure 4.6: Effect of Backoff Time in ABT/IT

the other hand, the number of retransmissions by the ABT layer must become large. In Fig. 4.6, we show the results of the ABT/IT case dependent on the mean backoff time. As shown in the figure, the throughput of long hop connections is still almost zero. It is because the effect of increasing the number of backoffs (by shorter backoff times) is only enjoyed by short hop connections.

We therefore take the second solution; enlarging the RTO value of the TCP in the case of TCP over ABT/IT. Figure 4.7 shows the result for this case. The RTO value of TCP is set to be ten times larger than the original one. Then, the throughput of the long hop connections can be very slightly improved at the expense of the decreased performance of short hop connections. The total throughput is also decreased; e.g., from 90.18 Mbps to 71.23 Mbps in the case of 1 msec propagation delay. Note that the tuning of TCP RTO value requires some changes of kernel sources in most operating systems.

Thus, we conclude that simple modifications of backoff times in ABT/IT and RTO values of TCP cannot resolve our "unfairness" problem. One solution is to

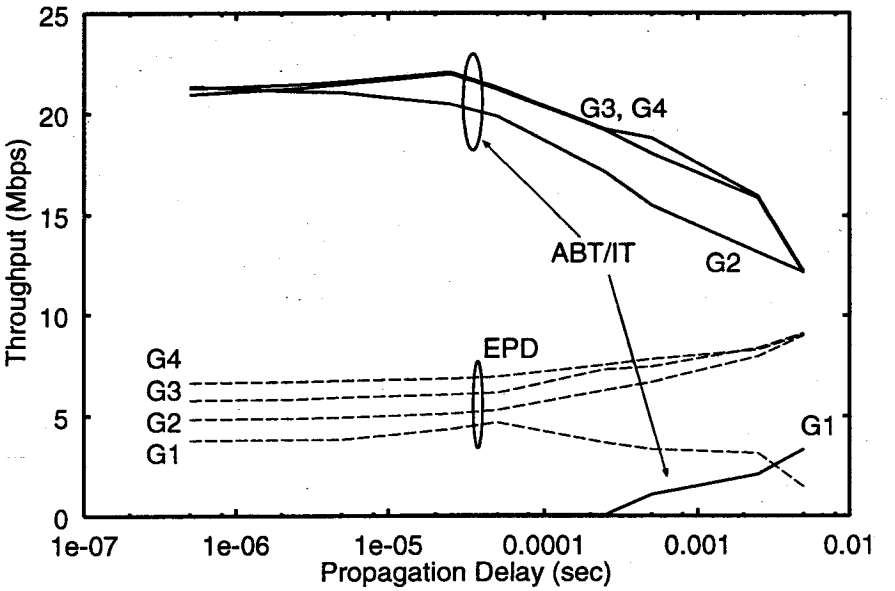


Figure 4.7: Mean Throughput Dependent on Propagation Delays in ABT/IT (Large RTO)

differentiate the backoff times dependent on the hop counts in ABT/IT. In Fig. 4.8, we show the simulation results. In obtaining the figure, we determine the backoff times of ABT dependent on the hop counts of connections; the backoff times of the short and long hop connections are set to be 92.8 msec and 928 μsec, respectively. The figure indicates that we can expect the performance improvement by carefully choosing the backoff times. However, it seems to be difficult since it heavily depends on other parameters, which includes the propagation delays as the figure shows.

We thus consider buffered ABT/IT, which will be shown in Subsection 4.2.3. Before doing so, we present the ABT/DT case in the next subsection.

4.2.2 Performance of TCP over ABT/DT

We first examine the basic performance of TCP over ABT/DT in Fig. 4.9 where TCP over EPD and TCP over ABT/DT are compared as a function of the propagation delay. TCP RTO threshold is not changed, and therefore the result corresponds to Fig. 4.8 for the ABT/IT case. The figure shows that the results are rather good if the

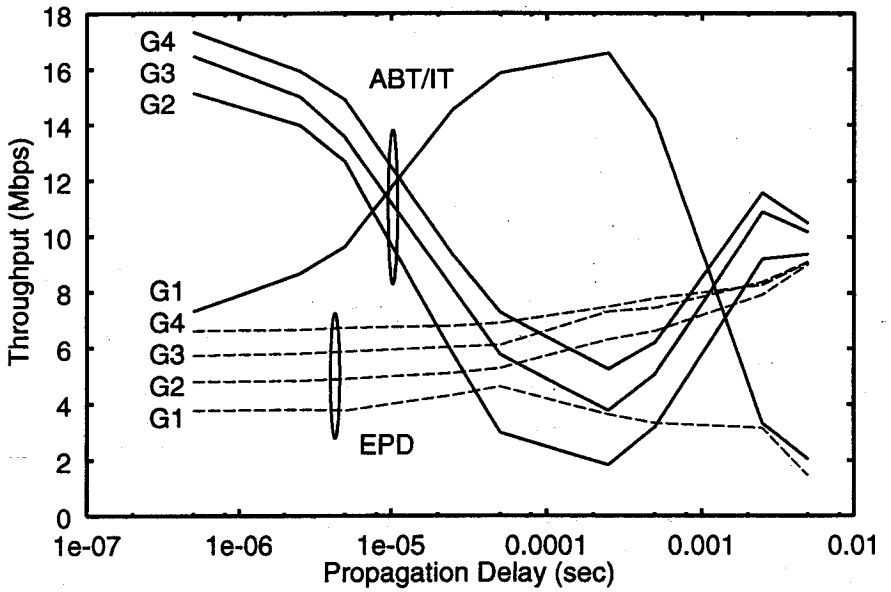


Figure 4.8: Mean Throughput Dependent on Propagation Delays in ABT/IT

propagation delay is small. When the propagation delay becomes long, however, the performance becomes suddenly degraded. As shown in [32, 31], this tendency that the performance of ABT/DT heavily depends on the propagation delay is a basic feature of ABT/DT since the burst transmission is allowed to be started only after the RM(ACK) cell is received by the sender.

However, the feature that the source starts burst transmission after it receives the RM(ACK) also makes it possible to introduce some extensions for the ABT/DT protocol. For example, ABT/DT with bandwidth negotiation which is investigated in [32, 31] is also applied to TCP network architecture. In ABT/DT, it is possible that the switch reserves the bandwidth less than the requested bandwidth specified in the RM cell if the requesting bandwidth is not available on the link. The switch then overwrites the new bandwidth on the RM cell to forward it to the next switch. In this bandwidth reservation method, each switch checks the bandwidth in the backward RM cell, and reduces the reserved bandwidth if the temporarily reserved bandwidth is larger than the one in the backward RM cell. The source receiving the backward RM(ACK) cell then starts to transmit the burst according to the bandwidth specified

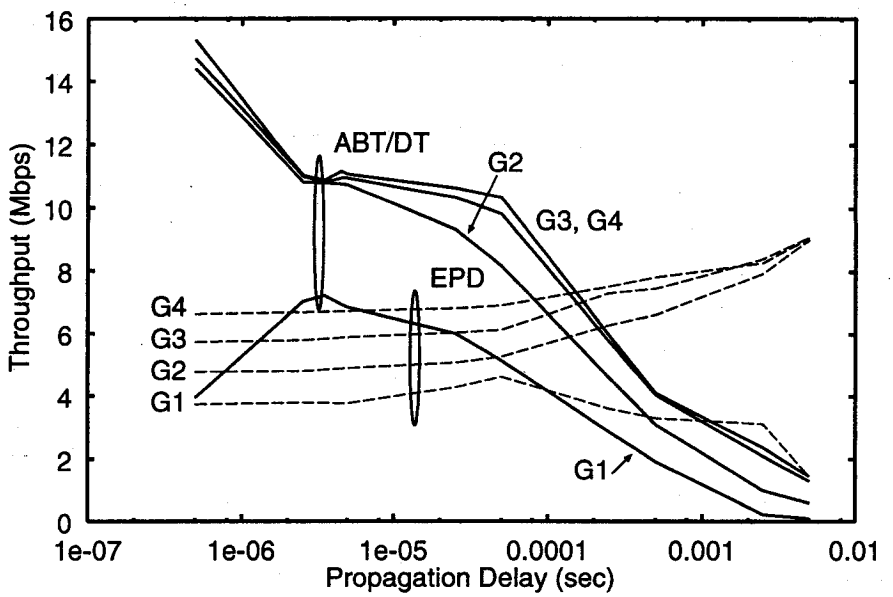


Figure 4.9: Comparisons between EPD and ABT/DT

in the RM cell.

We next show how much the performance can be improved by ABT/DT with bandwidth negotiation in Fig. 4.10. Other parameters are not changed from the previous case (Fig. 4.9). The throughput of ABT/DT with bandwidth negotiation is much increased in the short propagation delay case, and it is superior to the EPD case. The fairness between connections is also improved. Noting that we do not modify TCP RTO value in this case, we may conclude that ABT/DT is attractive for the case with short propagation delays.

4.2.3 Effects of Buffered ABT/IT to Support TCP

So far, we have not considered the buffer in the ABT protocols since the original ABT protocol only reserves the bandwidth in prior to the burst transmission. If we consider the buffer in addition to the bandwidth for reservation, much performance improvement can be expected. It is actually true as shown in [35]. In this subsection, we show the results by applying buffered ABT/IT to TCP.

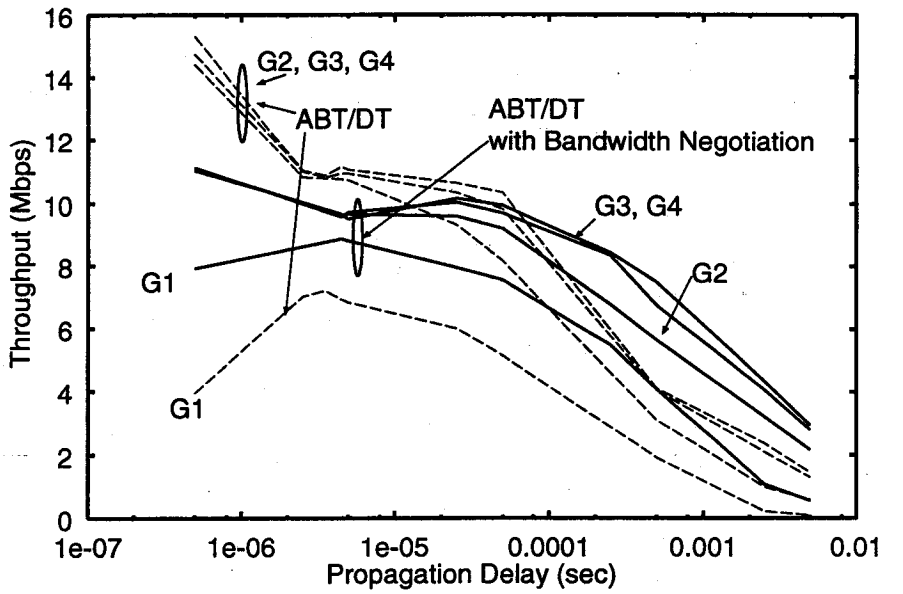


Figure 4.10: Effects of ABT/DT with Bandwidth Negotiation

We first examine the effect of the propagation delay in buffered ABT/IT. Figure 4.11 compares the throughput of TCP over buffered ABT/IT and TCP over EPD. It corresponds to Fig. 4.8 for the case of ABT/IT without buffering. That is, the TCP RTO value is not changed as an original value. However, the mean backoff times are changed according to hop counts of the connections. In contrast with the results presented in Subsection 4.2.1, the throughput of buffered ABT/IT becomes larger than the EPD case in all range of propagation delays. The difference between the throughput of long–hop connections and short–hop connections becomes small. Furthermore, we may say that the throughput performance is robust to the propagation delay in the sense that the throughput is not much affected by the propagation delay. It is very different from the previous case shown in Fig. 4.8. The main reason that we can establish such promising results is that by buffering bursts, the burst for long hop connections can also be served at the switches. If we allowed the modification of the TCP RTO value, the throughput is further improved. Figure 4.12 shows the case where the TCP RTO value is set to be ten times larger. In this case, buffered ABT/IT gives excellent results.

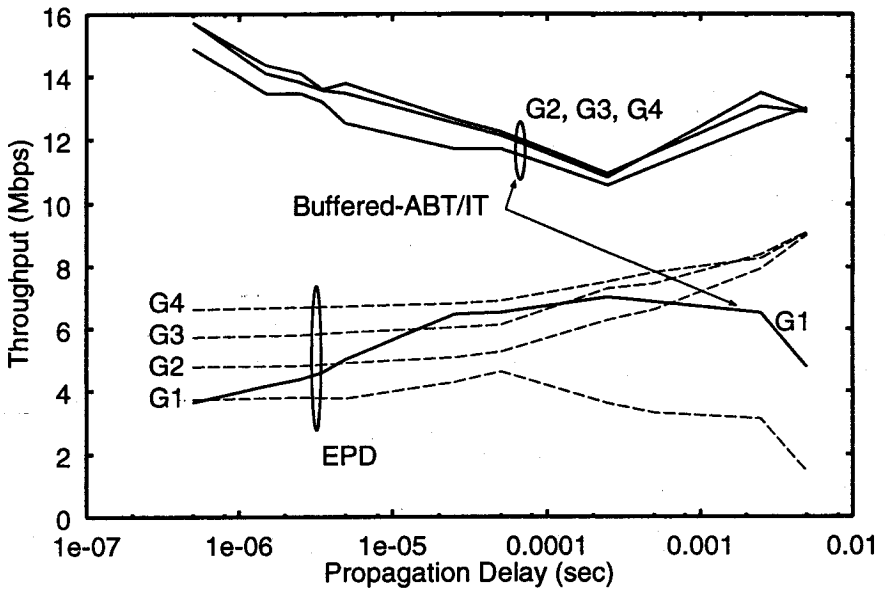


Figure 4.11: Mean Throughput Dependent on Propagation Delays (Buffered ABT/IT, EPD)

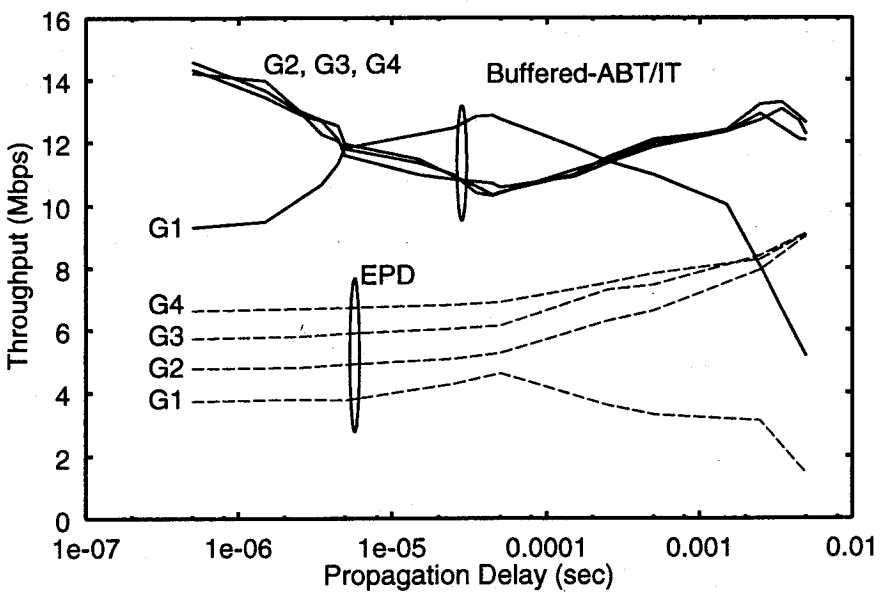


Figure 4.12: Mean Throughput Dependent on Propagation Delays (Buffered ABT/IT, EPD, Long RTO Case)

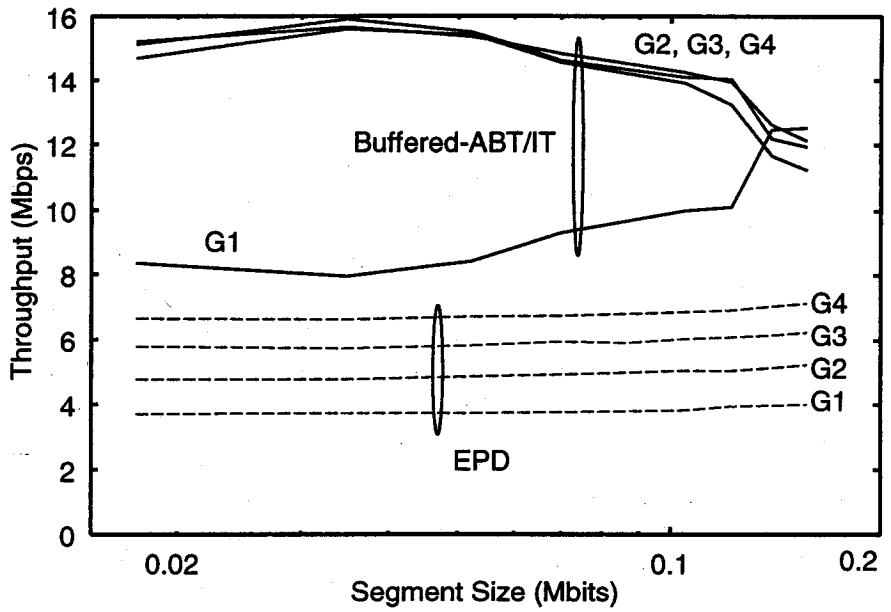


Figure 4.13: Effect of Segment Size on Throughput (Buffered ABT/IT)

The effectiveness of ABT can be attained as the burst size becomes large since the overhead due to the propagation delays can be eliminated. We last illustrate the influence of the segment size. Figure 4.13 shows the mean throughput against the segment size in two cases of buffered ABT and EPD. As shown in the figure, one may say that the throughput in the EPD case is not affected by the segment size owing to the wisdom of the EPD mechanism. On the other hand, the total throughput and fairness of the buffered ABT/IT can be improved as the larger segment size.

4.3 Concluding Remarks

In this chapter, we have investigated the performance of TCP over various ABT protocols. In such cases, it is necessary to take account of the fact that the congestion control is managed by the backoff mechanism of ABT service class and the window flow control of TCP. It is because those two congestion control mechanisms may interact with each other in an ill fashion, and the performance may be unexpected

degraded. Through simulation experiments, we have shown that retransmission mechanism (backoff) of ABT protocols can overlay the window flow control in TCP if the backoff times of ABT and retransmission timeout values of TCP are appropriately set. Otherwise, the total throughput can be larger than the TCP over EPD case, but the fairness among connections is lost. Among ABT protocols, buffered ABT/IT can offer good performance in terms of both of the throughput and fairness, and ABT/DT is the next if the propagation delay is small. Thus, ABT/DT with bandwidth negotiation is applicable to the LAN environment effectively.

the source and the switches are zero. It may not give fair comparisons since it is considered that the ABT protocols requires more processing overhead. By considering that the protocol processing delay is included in the propagation delay, the buffered ABT protocol still gives better performance than EPD. However, more investigation is required as a future research topic.

Chapter 5

Analysis of Network Traffic and its Application to Design of High–Speed Routers

In this chapter, we analyze the network traffic using the network traffic monitor and investigate the Internet traffic characteristics through a statistical analysis. We next show the application of our analytical results to parameter settings of high speed switching routers. Simulation results show that our approach makes highly utilized VC space and high performance in packet processing delay. We also show the effect of flow aggregation on MPLS. From our results, the flow aggregation has a great impact on the performance of MPLS.

5.1 Analysis of Traced Data

5.1.1 Analysis Approach

In this subsection, we introduce our analytic approach. We follow the statistical approach described in Paxson's previous work [23] where the approach was applied to the analysis of `telnet` and `ftp` traffics. The analysis of WWW traffic analyzed by

the same approach [24]. In the approach, we first choose several probability distributions, and determine parameters of those distribution functions based on traced data. In this chapter, we have adopted following distributions in addtion to an exponential distribution, an extreme distribution, a normal distribution. We consider a log–normal distribution and a log–extreme distribution. If the random variable $Y = \log X$ has a normal (extreme) distribution, then X is said to have a log–normal (log–extreme) distribution. Namely, log–normal and log–extreme distributions are defined as

$$F(x) = \int_0^x \frac{1}{\sqrt{2\pi}\sigma y} exp\Big[\frac{-(\log y - \zeta)^2}{2\sigma^2}\Big] dy, \tag{5.1}$$

and

$$F(x) = exp\Big[-exp\Big(-\frac{\log x - \alpha}{\beta}\Big)\Big]. \tag{5.2}$$

Those distributions were taken into account since they can cover a large range of values. We also consider a Pareto distribution which is defined as

$$F(x) = 1 - \Big(\frac{k}{x}\Big)^{\alpha}, \quad x \geq k, \tag{5.3}$$

Note that it is often used for modeling the self–similar traffics [25, 57].

We then test the goodness–of–fit of each model to select the most appropriate distribution via chi–squared examination. We use a criterion $\hat{\lambda^2}$ to choose the best model from the above–mentioned probability distributions. The criterion $\hat{\lambda^2}$ of each model is derived as follows. Suppose that we have observed n instances of random variables. We partition the range of those instance into N bins. Each bin has a probability p_i which is the proportion of the distribution falling into the ith bin. Let Y_i be the number of observation falling into the ith bin. Then $\hat{\lambda^2}$ is defined as

$$\hat{\lambda^2} = \frac{X^2 - K - N + 1}{n - 1}, \tag{5.4}$$

where

$$X^2 = \sum_{i=1}^{N} \frac{(Y_i - np_i)^2}{np_i}, \qquad K = \sum_{i=1}^{N} \frac{Y_i - np_i}{np_i}. \tag{5.5}$$

Finally, we choose the distribution with the smallest value of $\hat{\lambda^2}$ as a most accurate one.

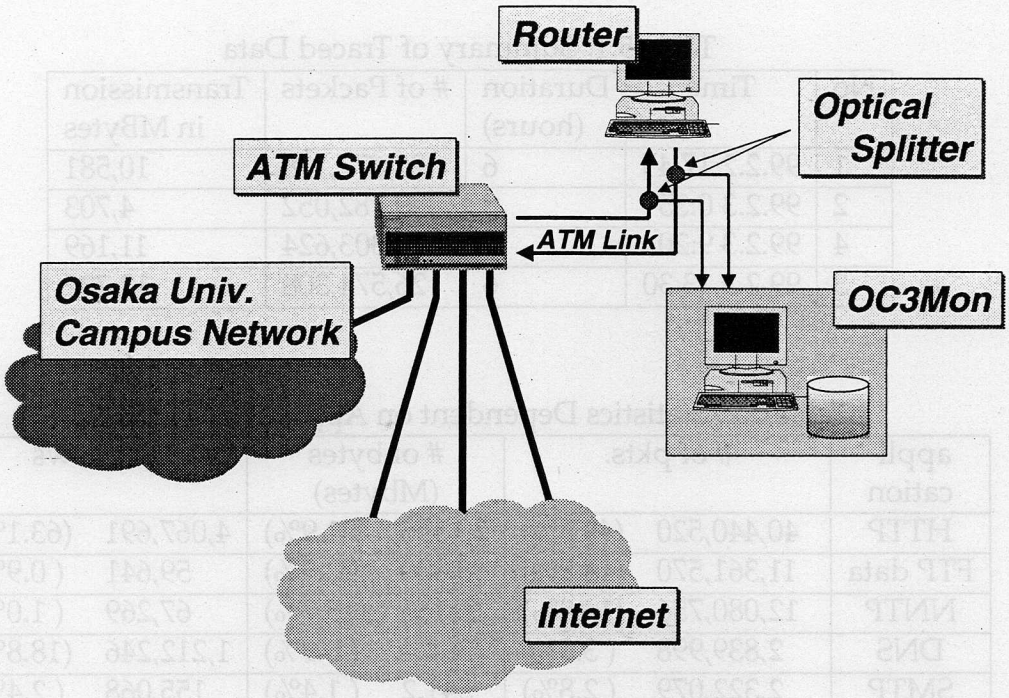


Figure 5.1: Configuration of OC3MON

5.1.2 Analytical Results

In this subsection, we give analytical results of the network traffic, which are gathered by the traffic monitor OC3MON. The monitor is placed at the gateway of Osaka University (see Figure 5.1), i.e., our results reflect the characteristics of the traffic between the Internet backbone and the large–scale local network.

Summary of the traced data collected by OC3MON is shown in Table 5.1. We monitored the gateway during a one day. Note that traced data is divided into four parts due to its file volume in Table 5.1. As shown in Table 5.1, the number of packets was 81,767,103 and the total transmission size was about 43.1 GBytes. Table 5.2 summarizes several statistics dependent on the application. We identified the packet stream having the same source address, destination address and application (port number) as an individual *flow*. As shown in the table, the ratio of HTTP traffic is very high and the volume of major three applications (HTTP, FTP, and NNTP) becomes over 90% of all traffic in bytes. It coincides recent trends of the network

Table 5.1: Summary of Traced Data

No	Time	Duration (hours)	# of Packets	Transmission in MBytes
1	99.2.2 17:45	6	22,077,118	10,581
2	99.2.3 0:35	8	9,182,052	4,703
4	99.2.3 9:20	4	23,303,624	11,169
3	99.2.3 13:30	6	26,574,308	12,737

Table 5.2: Statistics Dependent on Applications (ratio)

appli-cation	# of pkts.		# of bytes (Mbytes)		# of flows	
HTTP	40,440,520	(49.5%)	20,336	(51.9%)	4,067,691	(63.1%)
FTP data	11,361,570	(13.9%)	8,830	(22.5%)	59,641	(0.9%)
NNTP	12,080,756	(14.8%)	6,166	(15.7%)	67,269	(1.0%)
DNS	2,839,998	(3.5%)	394.4	(0.9%)	1,212,246	(18.8%)
SMTP	2,322,079	(2.8%)	541.2	(1.4%)	155,068	(2.4%)
FTP	611,559	(0.7%)	61.3	(0.2%)	33,968	(0.5%)
TELNET	1,010,214	(1.2%)	91.2	(0.2%)	50,895	(0.8%)
POP3	559,684	(0.7%)	119.6	(0.3%)	27,139	(0.4%)
others	10,540,732	(12.9%)	2,694	(6.9%)	770,370	(12.0%)

traffic reported in literatures [22, 58]. We can also see the statistical difference among applications. For example, the flow of FTP contains 15 packets in average while the DNS flow does only two packets.

Based on the analysis approach described in the previous subsections, we now present the statistical results.

- The distribution of IP address access frequencies

  The distribution of access frequencies of IP addresses is shown in Figure 5.2. The best model was a log–normal distribution. We can observe that the most of addresses are accessed at least twice, and that most frequently accessed WWW sites have more than 10,000 accesses.

- The distribution of the number of packets within the flow

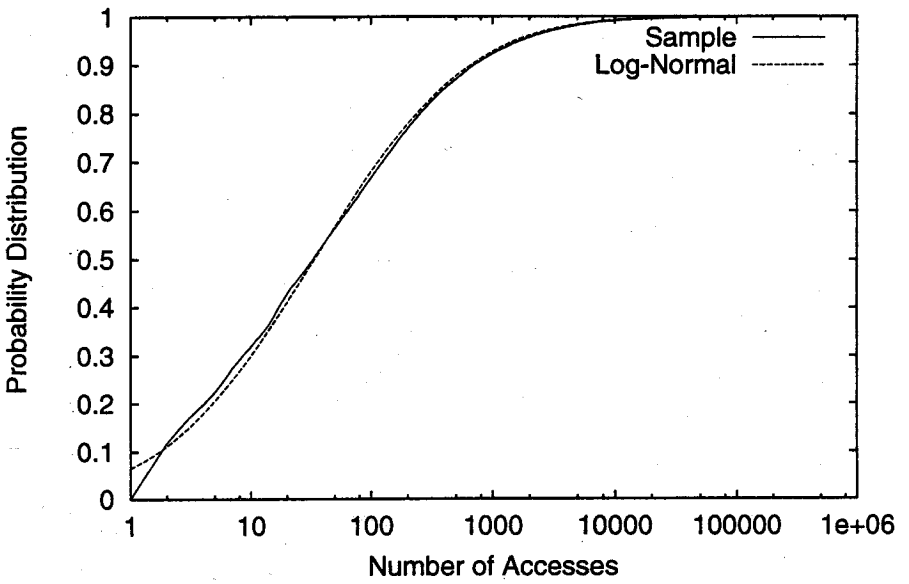


Figure 5.2: The Distribution of Access Frequencies of IP Addresses

We next show the analytic results of the distribution of the number of packets contained in each flow. Figure 5.3 shows the result. The best one was a log–normal distribution which has a long–tail.

However, if we focus on the tail of the distribution, the log–normal distribution cannot follow the traced lines [25]. Figure 5.4 shows the tail part of the distribution. As shown in the figure, a more suitable model is the Pareto distribution, which is known as a class of the heavy–tailed distribution decaying very slowly in its tail. It coincides recent studies on the traffic characterization on the Internet. However, we should note that the heavy-tailed distribution well fits only in its tail. If we consider the entire distribution, the log–normal distribution is best. Henceforth, we will consider the log–normal distribution for parameterizing the traffic flow in the next section.

- The distribution of flow duration

  The best model of the distribution of flow durations is also a log–normal distribution for the entire distribution, and a Pareto distribution for the tail. That

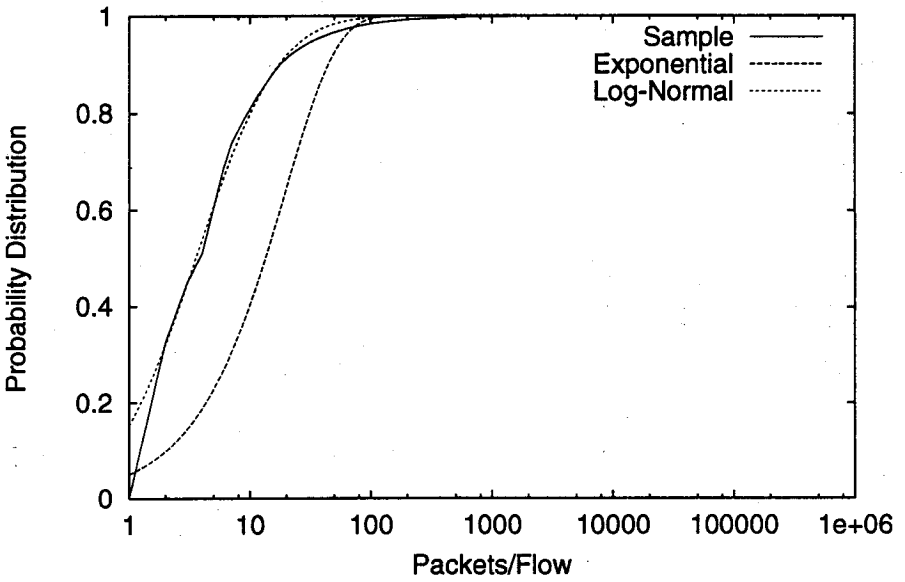


Figure 5.3: Distribution of the Number of Packets in Flows

is, the characteristics are same as the distribution of the number of packets in the flow described above. We omit the result due to space limitation.

- The distribution of flow intervals

  We finally show the inter-arrival distributions of flows in Figure 5.5. The best model is an exponential distribution. That is, we can assume that the flow arrivals follows a Poisson distribution.

5.2 Application to High Speed IP Switching

In this section we investigate the application of analytical results to determination of control parameters necessary in high speed switching routers. One important example is MPLS applied to ATM. In MPLS, the VC (virtual circuit) setting is activated by the predefined number X of packets contained in the flow, and it is released by the timeout value T. More specifically, when the number X of packets from the same flow are processed by the MPLS switch, the flow is recognized to need to set up

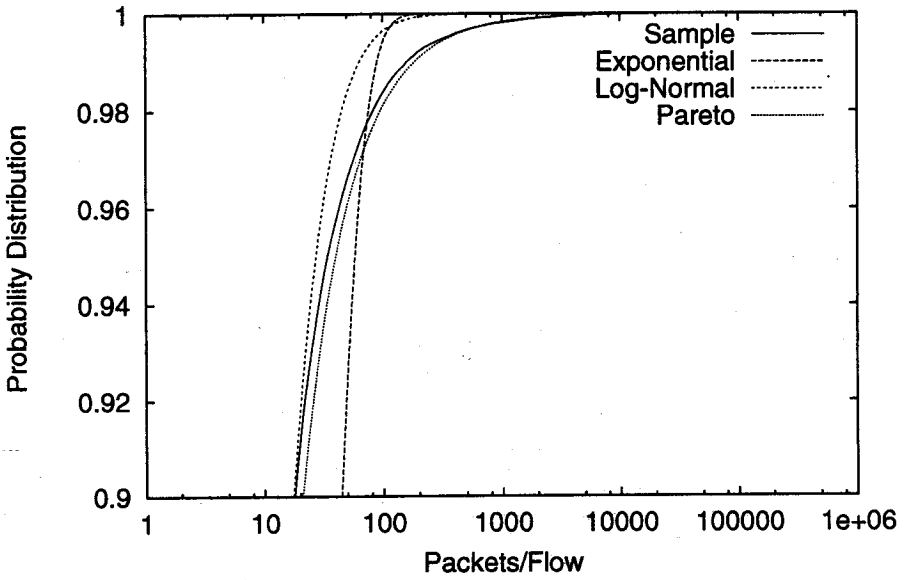


Figure 5.4: Distribution of the Number of Packets in Flows (Tail Part)

VC so that the faster hardware switching is performed. The assigned VC is released when the switch does not receive any packets of the flow within T seconds. Thus, the performance of MPLS is affected by the parameters X and T, which depends on the traffic characteristics of flows. If the parameter X is small, many VC assignments would be required not only for long–lived flows (which includes the large number of packets in the flow) but also for short–term flows, which results in the failure of setting more VCs for the long–lived flows newly arriving at the switch. On the other hand, if the parameter X is large, the utilization of VC space becomes low, and the switch performance is degraded. Moreover, the larger value of the parameter T leads to lower utilization of VC spaces while the switch with the small parameter T releases a VC of an active flow. Effect of a parameter set (X, T) in MPLS have been studied [27, 28], but those studies did not take account of traffic characteristics.

One difficult and unavoidable problem in determining appropriate values of X and T lies in that MPLS switch cannot isolate the flow if two or more consecutive flows have identical IP address and the port number. The timeout value T is then

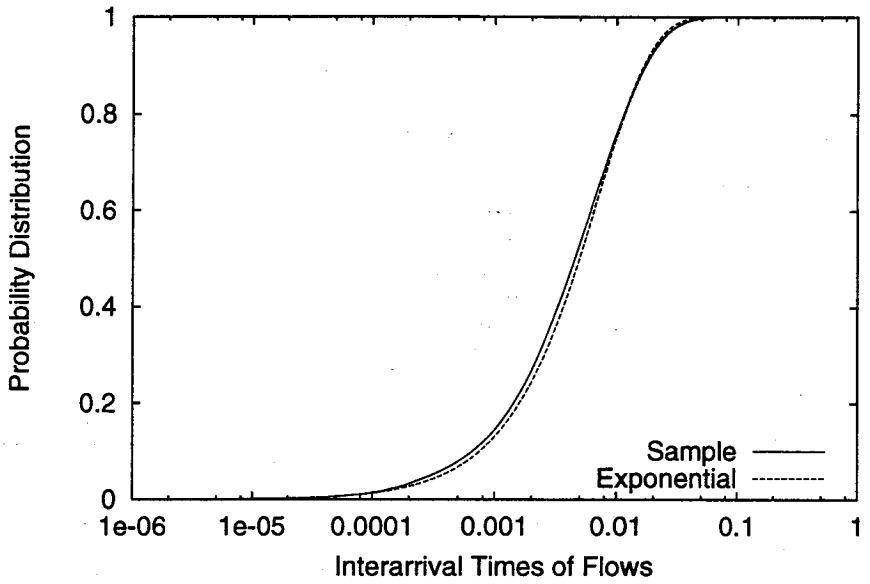


Figure 5.5: Distribution of Flow Inter-arrival Times

used to identify the flows. That is, the number of packets in the flow depends on the parameter T, and it is impossible to consider two parameters X and T independently. It is possible to obtain the appropriate value of T for each parameter X. However, it requires large computation time, and is not suitable for on–line calculations. In this section, we first demonstrate that the parameter X is a key for the switch performance. After that, the determination method of parameters X and T based on our statistical analysis. The effect of the parameter T is also discussed.

5.2.1 The Preliminary Results

While we have two parameters X and T for MPLS, we only consider X in this subsection to demonstrate that the parameter X plays an important role to achieve the high performance MPLS switch. In doing so, we assume that the switch can identify the end of the flow so that it immediately releases the VC setting at the end of flow. Of course, in the actual situation, it is not impossible from source/destination IP addresses and port numbers, unless the switch monitors, e.g., the "FIN" segment in

the case of TCP. We will discuss the effect of the parameter T in the next subsection.

To determine the appropriate parameter X, we introduce the following notations. Let $F(x)$ be the distribution function of the number of packets in flows (i.e., log–normal distribution according to our analysis) given by

$$F(x) = \int_0^x \frac{1}{\sqrt{2\pi}\sigma y} exp\left[\frac{-(\log y - \zeta)^2}{2\sigma^2}\right] dy. \tag{5.6}$$

Furthermore, $E_U(X)$ represents the mean number of packets of flows which contain less than X packets, and the mean processing time of the packet by "software" is denoted as δ. Similarly, $E_L(X)$ is the mean number of packets of flows having larger than or equal to X packets, and γ shows the packet processing delays in "hardware" switching.

We then have a mean flow processing time, Y, as

$$\begin{aligned} Y &= F(X)\delta E_U(X) && (5.7) \\ &\quad + (1 - F(X))\{\delta X + \gamma(E_L(X) - X)\}, && (5.8) \end{aligned}$$

Equation (5.8) shows that the router can process $1/Y$ flows in unit time. To process all flows without any packet losses, Y is required to satisfy $Y < 1/\lambda$ where λ is an arriving rate of flows. We then determine the value of X based on hardware specifications. From our traced data, we found that $X = 6$ was most appropriate. However, such a "static approach" does not take account of the fluctuation of the traffic load. The adaptive control method is thus necessary to effectively utilize the line capacity dependent on the traffic load. The approach of our adaptive control is next described.

We first introduce the time interval t_a. For each t_a, the switch observes the traffic load (the number of newly arriving flows), and changes the threshold value $X(t)$ adaptively. Then, we expect that the number of assigned VCs for every t_a is around the target value B. The target value B may be set by the static result, i.e.,

$$B = V_{max} - \lambda t_a(1 - F(X)). \tag{5.9}$$

We then determine the next $X(t + t_a)$ by balancing in and out flows for time interval t_a. For this, we introduce $V_d(t)$ as the variation caused by changing the parameter $X(t)$. We define $V_d(t)$ as

$$V_d(t) = t_a\lambda'(t)(1 - 2F(X(t_1)) + F(X(t))) \tag{5.10}$$

$$- V(t)R(t_a), \tag{5.11}$$

where $R(t_a)$ gives the ratio of the long–lived flows which ceases its connection within t_a. It is derived from the residual time for given distribution, i.e.,

$$R(t_a) = \frac{1}{\mu}\int^{t_a}(1 - F(X(t)))dt, \tag{5.12}$$

Then, the adequate threshold value for next t_a is determined by taking account of $V_d(t)$. That is, we calculate the smallest value of $X(t + t_a)$ satisfying

$$V_d(t) \leq B - V(X(t)). \tag{5.13}$$

Figure 5.6 shows the results of trace–drive simulation to compare the static and adaptive control methods. In simulation, the maximum number of VCs, V_{max}, is set to be 5000, and t_a is 2 sec. In the adaptive control, we set $B = 4,900$. As can be seen in the figure, VCs are highly utilized and its usage is stable around the threshold $B = 4,900$. Figure 5.7 shows the comparison of the mean packet processing times dependent on time. As shown in this figure, the benefit of highly utilized VCs leads to reduction of the packet processing time (83 μsec to 22 μsec).

5.2.2 Determination of Two Control Parameters

As having been demonstrated in the previous subsection, the parameter X is a key for the switch performance. However, we have to consider the parameter T in addition to the parameter X. As having been described before, the duration of the flow observed by the switch is affected by the parameter T.

Before describing the determination method, we first see its influence. For this purpose, we again analyzed the flow duration and the number of packets within

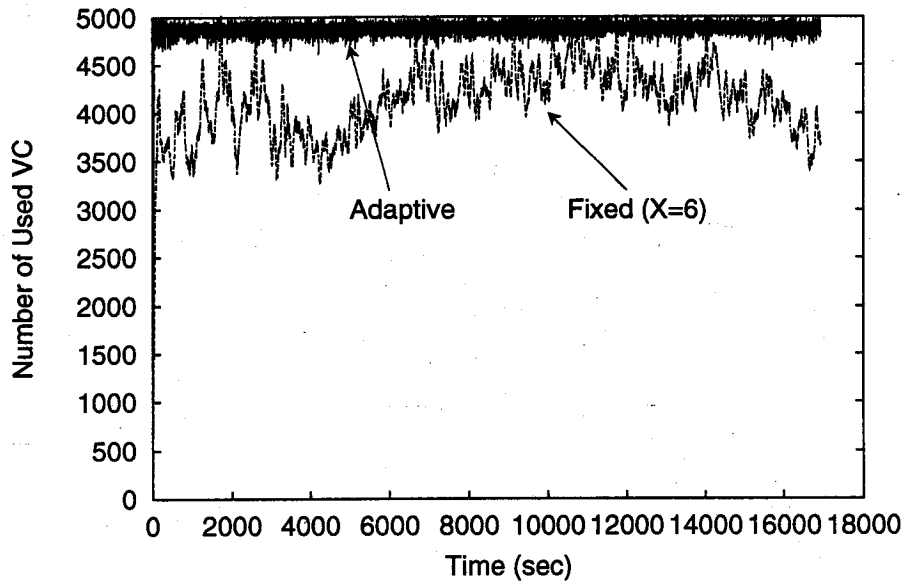


Figure 5.6: Comparison of the Numbers of Assigned VCs Dependent on Time

the flow from the traced data by varying the value of T. Note that the parameter X is assumed to be fixed at 5 during the experiments. Through the analysis, we confirmed that both of the flow duration and the number of packets within flows follow the log–normal distributions while the parameters of distributions depend on the parameter T. We also observed that the arrival process of flows follow the Poisson distribution.

Let S_T represent the random variable of the flow duration for given T. Since each flow holds the VC during $S_T + T$ and flows arrive according to the Poisson distribution, we may view the MPLS router as an $M/G/1/\infty$ queue by assuming that VCs are provided sufficiently. The steady–state probability p_j of our $M/G/\infty$ queue is obtained by

$$p_j = e^{-\lambda(E(S_T)+T)} \frac{\{\lambda(E(S_T)+T)\}^j}{j!}. \tag{5.14}$$

When the router can assign the number N_{VC} of VCs, the probability that VC assign-

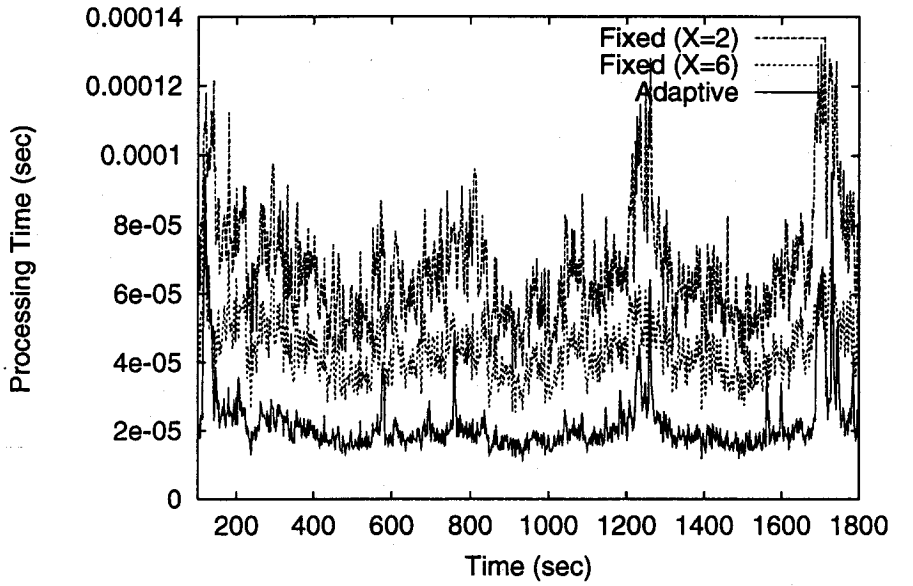


Figure 5.7: Comparison of Mean Packet Processing Times Dependent on Time

ment fails is approximately given by

$$L_{VC} = 1 - \sum_{j=0}^{N_{VC}} p_j. \tag{5.15}$$

and the average number of simultaneously assigned VCs, $\overline{N_{VC}}$, is given by

$$\overline{N_{VC}} = \sum_{j=1}^{N_{VC}} j p_j / (1 - L_{VC}). \tag{5.16}$$

From equation (5.15), we can calculate the minimum number of N_{VC} which is necessary to satisfy that L_{VC} is, e.g., less than 1%. Figure 5.8 shows such an example. In the figure, the relation between the parameter T and the minimum/average numbers of simultaneously assigned VCs. In plotting the figure, we set the parameter $X = 5$. From this figure, the number of VCs increases almost in proportion to the timeout value T. For instance, the appropriate value of T was 28 (sec) if the router has 5,000 VCs.

Remaining is that we need to know the statistics for various values of X, by which we can change the control parameters of MPLS routers according to the traffic load fluctuation. For this purpose, we first need to investigate the number of

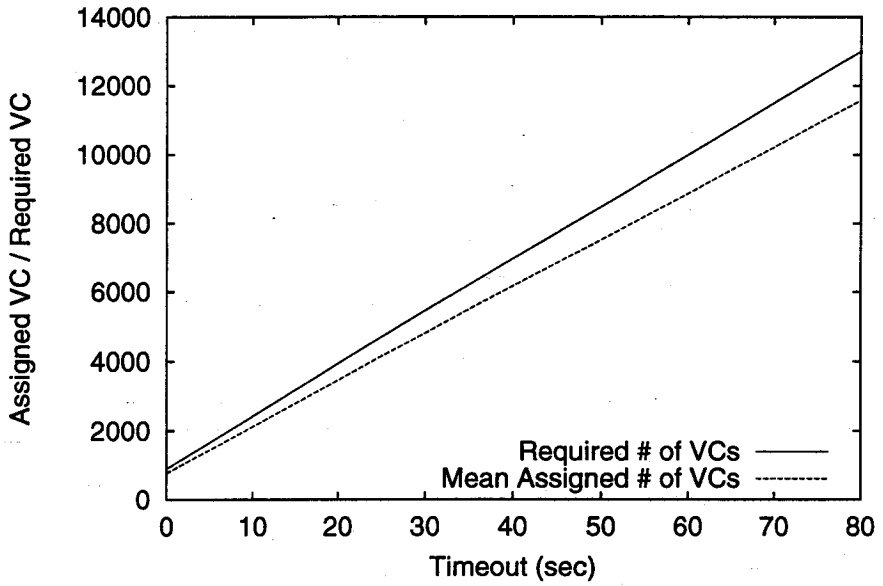


Figure 5.8: The Number of Simultaneously Assigned VCs ($X = 5$)

required VCs dependent on the parameter X (and T). We can determine it by examining the traced data for all possible values of X. However, it is apparently a time–consuming approach, and fortunately we also confirmed that the flow duration and the number of packets within the flow have a strong correlation.

We first consider the case of $X = 1$; i.e., the router assigns VCs to all flows. In this case, analytic results can be used directly. That is, the required number of VCs, N_{VC_1}, and the average number of VCs, $\overline{N_{VC_1}}$, are determined from equations (5.15) and (5.16).

For $X > 1$, VC assignment is performed when the Xth packet of the flow arrives at the router. The distribution of the holding time of the assigned VC $S(y)$ is given as

$$S(y) = \begin{cases} 0, & y < Z(X-1) \\ \frac{G(y-Z(X-1))}{(1-F(X))}, & \text{otherwise} \end{cases} \tag{5.17}$$

where $F(x)$ and $G(y)$ are distributions of the number of packet in flows and the flow duration, respectively. Note that both follow the log–normal distributions according

to our statistical analysis. Z is mean packet inter–arrivals of flows.

The arrival rate of flows λ_X for given X is

$$\lambda_X = (1 - F(X))\lambda \tag{5.18}$$

since VC is assigned only when the flow has a number X or more packets. By applying $S(y)$ and λ_X to an $M/G/\infty$ queue as in the previous subsection, we can determine the minimum number of VCs, N_{VC_X}, and the average number of VCs, $\overline{N_{VC_X}}$, from equations (5.15) and (5.16).

Finally, the processing load of the router, which is defined as ρ_{VC}, can be derived as

$$\rho_{VC} = \gamma\overline{N_{VC_X}} + \delta(\overline{N_{VC_1}} - \overline{N_{VC_X}}). \tag{5.19}$$

where the mean processing time of the packet by software is denoted as δ, and γ shows the packet processing delays in hardware switching.

Figure 5.9 shows the effect of the parameter X on the number of required VCs. In this case, we set T = 28 sec, $\gamma = 250$ μsec, $\delta = 10\mu$sec. As shown in this figure, $X = 3$ is an appropriate value when the VC space is 5,000. Figure 5.10 shows the effect of the parameter X on the processing load of the router. From this figure, $X \leq 6$ is necessary in order to be able to process all packets completely (i.e., $\rho_{VC} \leq 1$). This result coincides with the observation made in the previous subsection.

We finally show the effect of the parameter tunings in Figure 5.11. In the figure, we plot two cases of the timeout values; T = 60 sec and T = 30 sec. The parameter X was determined by the above equations. The VC space was set to be 5,000. As shown in the figure, we can observe that the mean packet processing time is decreased by tuning parameters, which leads to the performance improvement of the switching capacity of the router. We last note that since our formulation allows time–dependent arrival rate of flows (see equation (5.18)), the control parameters can be adaptive to the traffic load if the appropriate traffic load monitoring is performed.

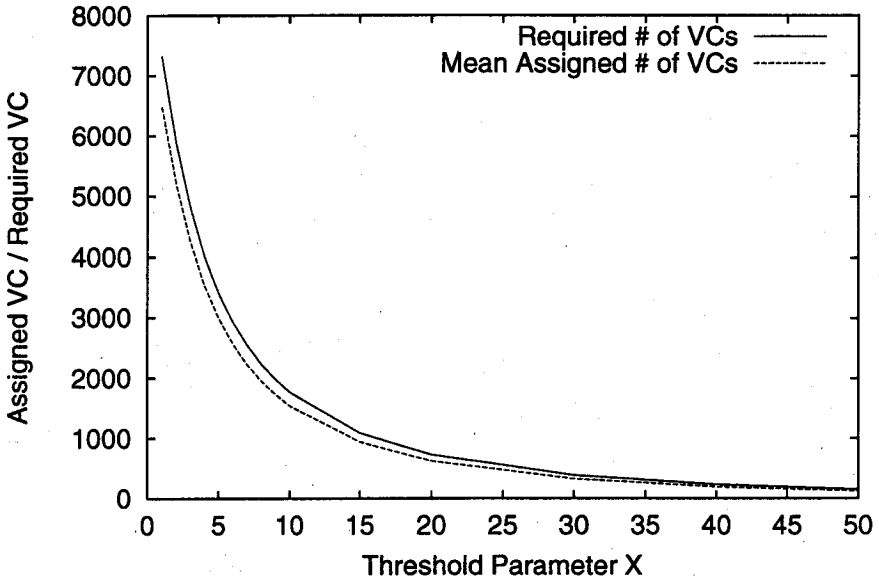


Figure 5.9: Relation between X and # of Assigned VCs

5.3 Effects of Flow Aggregation

So far, we assume that flows are identified by {*source host IP address, destination host IP address, source host port number, destination host port number*}. However, it is likely that the switching performance can be improved by flow aggregation, by which we mean that flows having the same destination port is treated by a single flow. As a result, it is expected that the router can assign VCs to more flows.

Such a performance improvement can be clearly expected when we consider the HTTP/1.0 protocol. Namely, when the WWW browser retrieves one HTML file and several in–line images, only one VC is required to transmit those files by the flow aggregation. Even when HTTP/1.1 is employed, an inappropriate setting of the timeout value T is likely to lead to the failure of performance improvement. It is because the MPLS router tends to treat the HTTP connection as multiple flows if the small value of the timeout T is selected.

In this subsection, we identify flows with three values

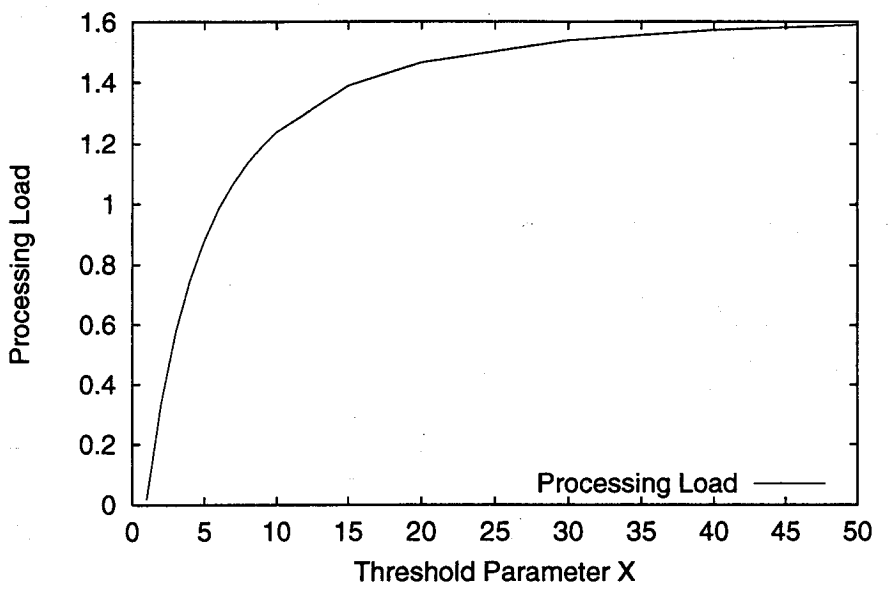


Figure 5.10: The Processing Load of the Router Dependent on X

Table 5.3: Result of Analysis of Aggregated Flows

Distribution	Distribution	Ratio
# of packets in the flow	log–normal	1.301
(Tail Part)	Pareto	
Flow duration	log–normal	1.096
(Tail Part)	Pareto	
Inter-arrivals of flows	log–normal	0.861
(Tail Part)	Pareto	

{*source host IP address, destination host IP address, destination host port number (application)*}

to aggregate flows, and show its effect on the performance of MPLS routers.

We first summarize the characteristics of aggregated flows in Table 5.3. The third column of the table (labeled by "Ratio") shows the ratio of aggregated flows to non–aggregated flows. For example, the average number of packets in aggregated flows is 1.301 times larger than the one in non–aggregated flows. From the table, it can be observed that durations of flows with and without aggregations are close. It is

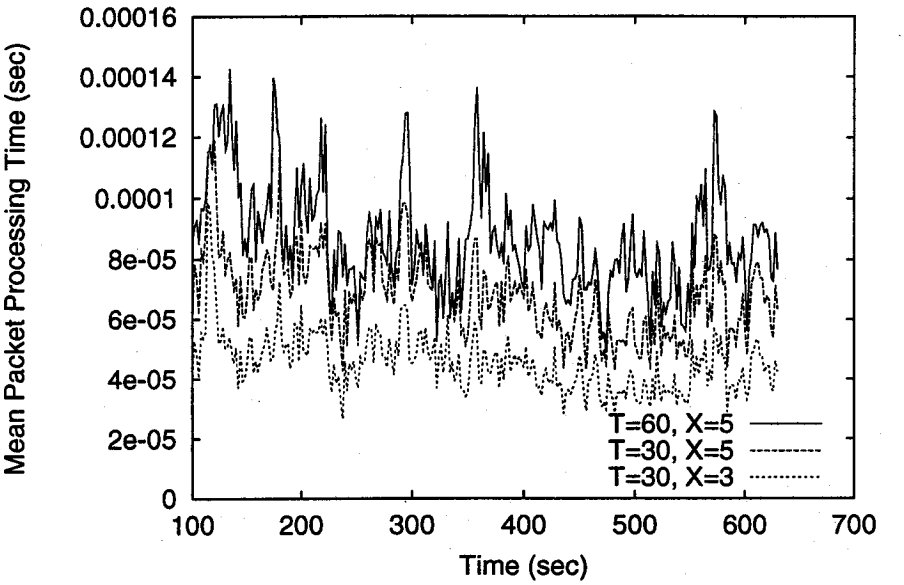


Figure 5.11: Effect of Tuning of Parameters X and T

because flow durations include the timeout value (T = 28 sec in the current case). If we exclude timeouts, the ratio becomes 1.75. Another observation in this table is that all of statistics have same distributions. Namely, we can apply our previous analysis to the case of aggregated flows.

To investigate the effect of flow aggregation on MPLS routers, trace–driven simulation was performed. Results on the mean packet processing time and the number of assigned VCs are shown in Figures 5.12 and 5.13, respectively. We determined the parameter X from Section 5.2.1. The number of simultaneously assigned VCs becomes drastically degraded (almost half) by flow aggregation. As a result, the mean packet processing delay becomes small and the router can assign VCs to more flows.

5.4 Concluding Remarks

In this chapter, we first give analytical results of the network traffic which are gathered by the traffic monitor. Through the statistical analysis, we found that most

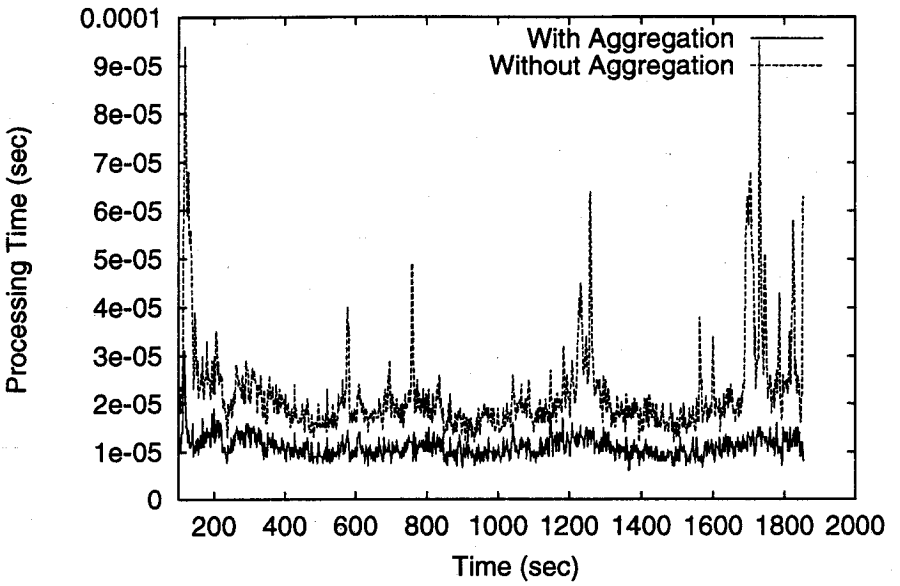


Figure 5.12: Effect of Flow Aggregation on Packet Processing Delays

statistics, including the number of packets in the flow and active flow durations, follow the log–normal distributions. We next investigate the application to parameter settings in high speed MPLS routers. From simulation results, VCs are able to be highly utilized and its usage is stable by applying the result of analysis for parameter settings. We also show the effect of flow aggregation. Simulation results show a clear performance improvement on number of used VCs.

In this chapter, we have not considered somce QoS levels dependent on applications. However, QoS level must be different among application, and therefore, the treatement of flows should be differentiated at the MPLS routers. For future research topics, it is necessary to consider some mechanism to application–dependent flow identification and VC setting.

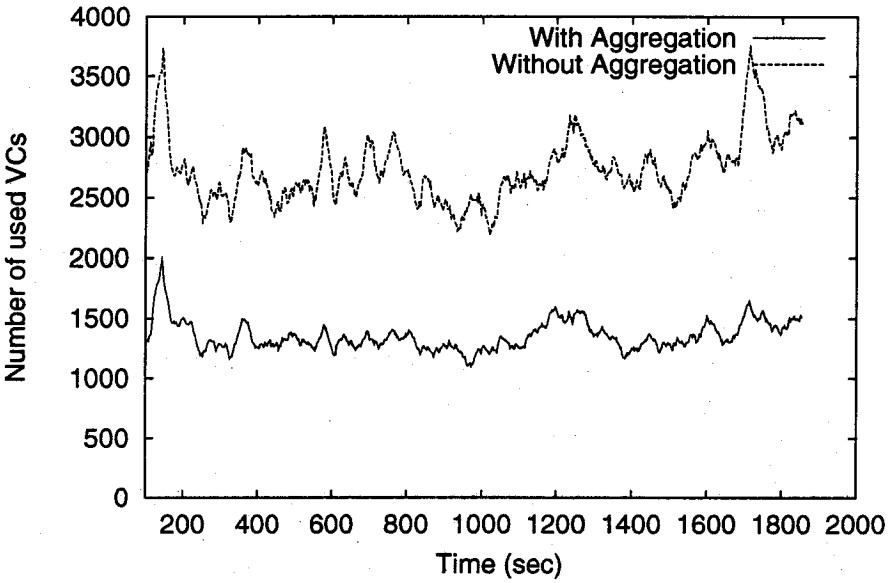


Figure 5.13: Effect of Flow Aggregation on the Rquired Number of VCs

Chapter 6

Conclusion

In this dissertation, we have investigated an effective data transfer mechanisms in high–speed networks. For this purpose, we have considered two issues; one is the traffic management in high–speed networks. The other is the new architecture of high-speed routers.

In Chapter 2, we have investigated the basic performance of ABT/IT and DT. Then, we have shown that ABT/IT is robust in the sense that its performance is not heavily affected by the propagation delay. On the other hand, ABT/DT is quite sensitive to the propagation delay. We next considered the performance improvement by the bandwidth negotiation mechanism which is only applicable to the ABT/DT protocol. Simulation results have shown that it is effective in the short propagation delay case if our concern is throughput, but the burst transmission delay is still larger than that of ABT/DT. We have also investigated effects of backoff methods to compare the burst transmission delays, and have observed similar tendencies as in the above cases. In the case of the short propagation delay, ABT/DT with bandwidth negotiation is most effective. When the propagation delay becomes large, on the other hand, ABT/IT with reduced bandwidth mechanism outperforms other methods.

In Chapter 3, we have proposed buffered ABT/IT and shown the effect of buffer reservation in ABT/IT. We have developed approximate analysis of buffered ABT/IT

to obtain the throughput and the mean transfer delay. Then, we have shown that buffer reservation in ABT/IT can lead to much performance improvement by comparing with the original ABT/IT protocol. We have also addressed the required buffer capacity for obtaining high throughput, and shown that it is a reasonable size based on the current switch technology. Finally, we have considered the bandwidth reduction mechanism that is helpful to improve the performance in the original ABT/IT protocol. However, in the case of the buffered–ABT/IT, it already provides high throughput and the performance improvement by the bandwidth reduction mechanism

In Chapter 4, we have investigated the performance of TCP over various ABT protocols. In such cases, it is necessary to take account of the fact that the congestion control is managed by the backoff mechanism of ABT service class and the window flow control of TCP. It is because those two congestion control mechanisms may interact with each other in an ill fashion, and the performance may be unexpected degraded. Through simulation experiments, we have shown that retransmission mechanism (backoff) of ABT protocols can overlay the window flow control in TCP if the backoff times of ABT and retransmission timeout values of TCP are appropriately set. Otherwise, the total throughput can be larger than the TCP over EPD case, but the fairness among connections is lost. Among ABT protocols, buffered ABT/IT can offer good performance in terms of both of the throughput and fairness, and ABT/DT is the next if the propagation delay is small. Thus, ABT/DT with bandwidth negotiation is applicable to the LAN environment effectively.

In Chapter 5, we have analyzed the actual network traffic gathered by the traffic monitor. Through the statistical analysis, we have shown that the number of packets in the flow, and active flow durations follow the log–normal distributions whose tail follows the pareto distribution, which is known as a class of the heavy–tailed distribution. We have next determined parameters in high–speed MPLS routers based on those observations. It is shown that VCs are stably and highly utilized by applying the analytical results on traffic characteristics. We have also shown the effect of flow

aggregation, in which flows are aggregated into one with a larger granularity of classification. These properties give a significant impact on the performance of MPLS routers. Simulation results show a clear performance improvement on number of used VCs.

Bibliography

[1] ITU-T, "Traffic control and congestion control in B-ISDN," *Recommendation I.371.*

[2] The ATM Forum, "Traffic management specification version 4.0," *ATM Forum Contribution 95-0013R9*, December 1995.

[3] M. W. Garrett, "A service architecture for ATM: From applications to scheduling," *IEEE Network*, vol. 10, pp. 6–14, May/June 1996.

[4] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Rate-based congestion control for ATM networks," *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 60–72, April 1995.

[5] P. E.Boyer and D. P.Tranchier, "A reservation principle with applications to the ATM traffic control," *Computer Networks and ISDN Systems*, vol. 24, pp. 321–334, 1992.

[6] F. Guillemin and P. Boyer, "ATM block transfer capabilities: The special case of ABT/DT," in *Proceedings of IEEE GLOBECOM '96*, pp. 762–766, November 1996.

[7] H. Suzuki and F.A. Tobagi, "Fast bandwidth reservation scheme with multi–line & multi–path routing in ATM networks," in *Proceedings of IEEE INFOCOM '92*, pp. 2233–2240, 1992.

[8] J. Enssle, U. Briem, and H. Kröner, "Performance analysis of fast reservation protocols for ATM," *Proceedings of IFIP 2nd Workshop on Performance Modelling and Evaluation of ATM Networks*, July 1994.

[9] L. Cerda, J. Garcia, and O. Casals, "A study of the fairness of the fast reservation protocol," *Proceedings of IFIP 3rd Workshop on Performance Modelling and Evaluation of ATM Networks*, July 1995.

[10] D. P.Tranchier, P. E.Boyer, Y. M.Rouaud, and J. Mazeas, "Fast bandwidth allocation in ATM networks," *Proceedings of ISS '92*, vol. 2, pp. 7–11, Octber 1992.

[11] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols.* Reading, Massachusetts: Addison-Wesley, 1994.

[12] E. C. Rosen, A. V. Than, and R. Callon, "Multiprotocol label switching architecture," *IETF Internet Draft*, Mar. 1998.

[13] I. Widjaja, "Random access for ATM LANs and WANs," in *Proceedings of IEEE ICC '94*, pp. 39–43, 1994.

[14] S. S. Lam and G. G. Xie, "Burst scheduling: Architecture and algorithm for switching packet video," in *Proceedings of IEEE INFOCOM '95*, pp. 940–950, 1995.

[15] S. S. Lam and G. G. Xie, "Group priority scheduling," in *Proceedings of IEEE INFOCOM '96*, pp. 1346–1356, 1996.

[16] G. G. Xie and S. S. Lam, "Real–time block transfer under a link sharing hierarchy," in *Proceedings of IEEE INFOCOM '97*, pp. 388–397, 1997.

[17] H. Shimonishi, T. Takine, M. Murata, and H. Miyahara, "Performance analysis of fast reservation protocols in ATM networks," *Performance Evaluation*, vol. 26, pp. 263–287, 1996.

[18] H. Shimonishi, T. Takine, M. Murata, and H. Miyahara, "Performance analysis of fast reservation protocol with generalized bandwidth reservation method," in *Proceedings of IEEE INFOCOM '96*, pp. 758–767, 1996.

[19] H. Shimonishi, T. Takine, M. Murata, and H. Miyahara, "Performance analysis of fast reservation protocol in ATM networks with arbitrary topologies," *Performance Evaluation*, vol. 27&28, pp. 41–69, 1996.

[20] B. T. Doshi and H. Heffes, "Performance of an in-call buffer-window reservation/allocation scheme for long file transfers," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1013–1023, September 1991.

[21] H. T. Kung and A. Chapman, "Credit-based flow control for ATM networks: Credit update protocol, adaptive credit allocation, and statistical multiplexing," in *Proceedings of ACM SIGCOMM '94*, pp. 101–114, October 1994.

[22] K. Thompson, G. J. Miller, and R. Wilder, "Wide–area Internet traffic patterns and characteristics," *IEEE Network*, pp. 10–23, November 1997.

[23] V. Paxson and S. Floyd, "The failure of Poisson modeling," in *Proceedings of ACM SIGCOMM '94*, pp. 257–268, 1994.

[24] M. Nabe, M. Murata, and H. Miyahara, "Analysis and modeling of WWW traffic for capacity dimensioning for Internet access lines," in *Proceedings of SPIE Synposium on Performance & Control of Network Systems*, November 1997.

[25] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic evidence and possible causes," in *Proceedings of ACM SIGMETRICS '96*, pp. 160–169, 1996.

[26] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carring web traffic over flow–switched networks," *IEEE/ACM Transactions on Networking*, pp. 673–685, December 1998.

[27] S. Lin and N. McKeown, "A simulation study of IP switching," in *Proceedings of ACM SIGCOMM '97*, pp. 15–24, 1997.

[28] H. Che and S. Q. Li, "Adaptive resource management for flow-based IP/ATM switching systems," in *Proceedings of IEEE INFOCOM '98*, pp. 381–389, April 1998.

[29] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance comparisons of ABT/IT and DT in ATM networks," in *Proceedings of Symposium on Performance Models for Information Communication Networks '96*, pp. 326–337 (in Japanese), January 1997.

[30] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance comparisons of ABT/IT and DT in ATM networks," *IEICE Technical Report (IN96-125)*, pp. 121–128 (in Japanese), January 1997.

[31] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance comparisons of ABT/IT and DT in ATM networks," in *Proceedings of IEEE GLOBECOM '97*, pp. 1361–1366, November 1997.

[32] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance comparisons of ABT/IT and DT in ATM networks," *Journal of Operations Research Society of Japan*, vol. 41, pp. 35–53, March 1998.

[33] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance improvement of ABT protocols with combined bandwidth/buffer reservation," *IEICE Technical Report (IN97-104)*, pp. 37–42 (in Japanese), Sepetmber 1997.

[34] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance improvement of ABT protocols with combined bandwidth/buffer reservation," in *Proceedings of Symposium on Performance Models for Information Communication Networks '97*, pp. 284–293, January 1998.

[35] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance improvement of ABT protocols with combined bandwidth / buffer reservation," in *Proceedings of IEEE ATM '98 Workshop*, pp. 129–136, May 1998.

[36] S. Ata, T. Takine, M. Murata, and H. Miyahara, "Performance improvement of ABT protocols with combined bandwidth / buffer reservation," submitted to *Performance Evaluation*, 1998.

[37] S. Ata, M. Murata, and H. Miyahara, "Performance evaluation of TCP over ABT protocols," *IEICE Technical Report (SSE97–198)*, pp. 31–36 (in Japanese), March 1998.

[38] S. Ata, M. Murata, and H. Miyahara, "Performance evaluation of TCP over ABT protocols," in *Proceedings of SPIE Symposium on Voice, Video, and Data Communications*, pp. 423–433, November 1998.

[39] S. Ata, M. Murata, and H. Miyahara, "Performance evaluation of TCP over ABT protocols," *IEICE Transactions on Communications (B-I)*, pp. 951–959, May 1999.

[40] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic for the design of high–speed layer 3 switches," *IEICE Technical Report (CQ98–35)*, pp. 29–36 (in Japanese), September 1998.

[41] S. Ata, M. Murata, and H. Miyahara, "Application of network traffic analysis for the design of high–speed routers," in *Proceedings of IEICE Telecommunication Management Workshop*, vol. J82–B, pp. 81–86 (in Japanese), March 1999.

[42] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic and its application to design of high–speed routers," in *Proceedings of ITC–CSCC '99*, pp. 788–791, July 1999.

[43] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic and its application to design of high–speed routers," in *Proceedings of SPIE Symposium on Internet II: Quality of Service and Future Directions*, pp. 423–433, September 1999.

[44] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic and its application to design of high–speed routers," to appear in *IEICE Transactions on Information and Systems (D-I)*, May 2000.

[45] S.-P. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing," *IEEE/ACM Transactions on Networks*, vol. 1, pp. 105–112, 1993.

[46] F. P. Kelly, "Blocking probabilities in large circuit-switched networks," *Advances in Applied Probability*, vol. 18, pp. 473–505, 1986.

[47] F. P. Kelly, "Routing in circuit-switched networks: Optimization, shadow prices and decentralization," *Advances in Applied Probability*, vol. 20, pp. 112–144, 1988.

[48] F. P. Kelly, "Routing and capacity allocation in networks with trunk reservation," *Mathematics of Operations Research*, vol. 15, pp. 771–793, 1990.

[49] F. P. Kelly, "Loss network," *Annals of Applied Probability*, vol. 1, pp. 319–378, 1991.

[50] W. Whitt, "Blocking when service is required from several facilities simultaneously," *AT&T Technical Journal*, vol. 64, pp. 1807–1856, 1985.

[51] F. P. Kelly, "Reversibility and stochastic networks," *Jon Wiley & Sons, Chichester*, 1979.

[52] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, December 1988.

[53] M. Ilas and H. Mouftah, "Quasi cut-through: A new hybrid switching technique for computer communication networks," *IEE Proceedings*, vol. 131 Pt.E, pp. 1–9, Jan 1984.

[54] A. Abo-Taleb and H. Mouftah, "Delay analysis for interfered paths under general cut-through switching," *Canadian Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 80–84, 1988.

[55] Q. Ma and P. Steenkiste, "Quality-of-service routing for traffic with performance guarantees," in *Proceedings of IFIP IWQOS '97*, pp. 115–126, May 1997.

[56] A. Romanow and S. Floyd, "Dynamics of TCP over ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 633–641, May 1995.

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

[58] G. J. Miller and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings of INET '98 avaiable at* `http://www.isoc.org/inet98/proceedings/6g/6g_3.htm`, April 1998.