Regulation of TCP Flow in Heterogeneous Networks Using Packet Discarding Schemes

by

Yu-Shiou Flora Sun

B.S. Electrical Engineering and Computer Science (1997)

Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the Requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology

January 16, 1998

Author .....
Department of Electrical Engineering and Computer Science
February 9, 1998

Certified by .....
Kai-Yeung (Sunny) Siu
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses
Department of Electrical Engineering and Computer Science

Eng.

Regulation of TCP Flow in Heterogeneous Networks Using Packet Discarding Schemes

by

Yu-Shiou Flora Sun

Submitted to the
Department of Electrical Engineering and Computer Science

January 16, 1998

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Flow control in heterogeneous networks requires careful planning, especially in the case of TCP over ATM, because TCP and ATM each protocol has a different flow control mechanism. The early packet discard scheme (EPD) is commonly used with ATM's unspecified bit rate service (UBR). This thesis presents an improvement to EPD called single packet discard scheme (SPD). The new scheme drops a packet early in the congestion state, so TCP can slow down its rate before the congestion becomes more serious, which can lead to buffer overflow and packets loss. With the SPD scheme, the end-to-end goodput (effective throughput) to input ratio is improved so that a more efficient use of the bandwidth can be reached. If less bandwidth is wasted due to congestion, the preserved bandwidth can be used by other applications and result in a better overall bandwidth utilization. For a configuration used in exploring the goodput performance affected by the SPD scheme, our preliminary result shows an average of 13% improvement for the overall goodput measurement.

Thesis Supervisor: Kai-Yeung (Sunny) Siu
Title: Assistant Professor of Mechanical Engineering

Acknowledgment

I would like to take the time to thank several people whose support made this thesis possible. First, I would like to thank Professor Sunny Siu for the opportunity to work on this project and for his guidance and visions which taught me what research was all about.

Next, I would like to thank Dr. Wenge Ren for helping me getting start on NetSim. A special thanks to my other mentor Yuan Wu for his support in many steps of the thesis. I would like to acknowledge that the simulation settings in section 5.6 was a result of collaboration with him. I would also like to thank my lab colleagues Anthony, Paolo and Wayne who were always willing to help me when they could.

Special thanks to Grant, Alice and Ien for their help in proofreading this thesis. I would like to thank Yi-Mei, Pei-Ting, Roxy, Yi-San, Ginger, Cynthia, Sunny, and Henry for their constant encouragements. Thanks to Mrs. Liao's daily home-made meals during the last few weeks when I was trying to finish the thesis.

I would like to thank my best friends, Jung-Chi, Chunyi, and Jung-Sheng for their supports have always been invaluable to me.

Finally, I would like to dedicated this thesis to my parents whose love kept me going. Thank you for always believed in me.

Table of Contents

1 Introduction 6
1.1 Motivation 6
1.2 Contribution of this Thesis 7
1.3 Overview 8
2 Flow Control Mechanisms: an Overview 10
2.1 TCP Flow Control -- Adaptive Retransmission Mechanisms 10
2.1.1.Retransmission Timer and Backoff Mechanism 10
2.1.2 Window-based Flow Control 11
2.1.3 Maximum Segment Size Computation 11
2.1.4 Congestion Avoidance and Control 11
2.1.5 Round-trip Delay Estimation 12
2.2 The ATM Traffic Management and the Five Services Categories 12
2.2.1 The Constant Bit Rate Service 12
2.2.2 The Real-Time Variable Bit Rate Service 13
2.2.3 Non-Realtime Variable Bit Rate Service 13
2.2.4 Unspecified Bit Rate Service 13
2.2.5 Available Bit Rate Service 14
2.3 Internetworking of TCP over ATM 14
2.3.1 ABR versus UBR in Flow Control 15
2.3.2 TCP/IP over UBR Service 15
2.3.3 TCP over ABR Service 16
2.4 Previous Research 17
2.5 Existing Early Packet Discard Schemes 19
3 The Proposed Single Packet Drop Scheme 21
3.1 Motivation 21
3.2 Description of the Single Packet Discard Scheme 21
3.3 The Implementation of Single Packet Discard Mechanism 23
4 Simulation Setup 25
4.1 Network Simulator (NetSim) 26
4.1.1 User Component 28
4.1.2 The TCP Components 29
4.1.3 The Physical Links 29
4.1.4 The ATM Edge Device 30
4.1.5 The ATM Switches 30
4.2 An Explanation on Goodput 30
4.3 Four Goals 31
5 Simulation Results and Analysis 32
5.1 Overview
5.2 The Dropping Threshold of SPD 32
5.3 Performance Comparison of EPDs and OSD for the Persistent Sources 46
5.4 End-to-end Performance for Bursty Data Source 48
5.5 Comparing the Switch Queue for EPD and SPD schemes 50
5.6 Overall Goodput Improvement 50

6 Concluding Remarks 54
6.1 Summary 54
6.2 Future Works 55
Bibliography 56

Chapter 1

Introduction

1.1 Motivation

Flow control in heterogeneous networks requires careful planning, especially for networks composed of subnetworks, each with differently-designed flow control mechanisms. Within a TCP/IP over ATM network, ATM control mechanisms are only applicable to the ATM subnetwork, while the TCP flow control extends from end to end. One solution is to employ the flow control algorithm that works for the end-to-end ATM network in the ATM subnetwork, but this may actually worsen the performance of the overall TCP network[1].

One commonly used end-to-end ATM flow control is the available bit rate service (ABR). The ABR flow control mechanisms regulate the input rate into the ATM subnetwork according to the congestion condition within the subnetwork. Although ABR can successfully keep the ATM subnetwork free of congestion by slowing down the traffic, it simply pushes and postpones the congestion to the edge of the subnetwork. Moreover, ABR fails to immediately inform TCP with the congestion status, and TCP keeps sending data into the already congested ATM network. Finally, the complexity of ABR incurs significant overhead for implementing ABR switches. Therefore, other schemes are proposed to deal with this TCP over ATM problem without employing the ABR service.

The unspecified bit rate service (UBR), which allows TCP connections to transmit information without much overhead, is also commonly used. Rather than actively controlling the input rates into the ATM subnetwork, like ABR service, the UBR service simply allows switch buffers to build up and drop incoming cells if the buffers overflow. If a cell is dropped, TCP slows down its rate into the ATM subnetwork to cope with the congestion. The advantage of UBR is that it does not further postpone the time for TCP to slow

down when congestion occurs in the ATM subnetworks. However, it is still undesirable for the UBR service to just drop many cells during congestion, since data retransmission increases with the number of dropped packets. This induces a vicious positive feedback of retransmission, more congestion, and more packet loss.

There are two existing schemes for addressing this problem: the partial packet discard scheme (PPD) and early packet discard scheme (EPD)[7]. Both of these schemes are ATM cell-discarding techniques which maximize the system's effective throughput (goodput) by taking advantage of the fact that ATM traffic is made up of large TCP packets which are segmented into a series of ATM cells. If any cells of a packet are dropped, there is no need to send the other cells belonging to the same packet since the entire packet will have to be retransmitted by the higher level protocol. The act of discarding all the remaining cells is called the partial packet discard scheme (PPD). The early packet discard scheme(EPD) is used when cells are dropped because all the cells belonging to the packet might not be able to fit into the available switch buffer space. The difference between these two schemes is that EPD acts before the cells are admitted to the output buffer and PPD acts after the cells have been admitted to that buffer.

During the congestion, EPD scheme drops cells if the switch buffer size reaches beyond the EPD thresholds. In general, the EPD threshold is set somewhere between 50% to 100% of the whole of the maximum buffer size. When the size of the switch buffers reach the EPD threshold, the congestion is already very severe. In a majority of cases, EPD drops all of the ATM cells that make up multiple TCP packets. TCP eventually realizes that packets are lost, but by then the network congestion has increased.

1.2 Contribution of this Thesis

Even with EPD and PPD, employing UBR service in the TCP over ATM networks may

still yield poor performance. This thesis introduces a modification to EPD, called the single packet discard scheme (SPD), that increases the end-to-end performance of TCP over ATM heterogeneous networks. The performance is measured in terms of the goodput to input ratio of the end-to-end traffic.[1] Goodput to input ratio signifies bandwidth efficiency. If less bandwidth is wasted due to congestion, the preserved bandwidth can be better used by other applications. With the new single packet discard scheme (SPD), TCP is notified much earlier and congestion is relieved much faster.

SPD causes TCP to slow down the input rate quickly after congestion occurs. The input is reduced to slightly above the utilizable bandwidth during congestion which results in high goodput to input ratio. SPD saves the bandwidth in two ways: (1) it preserves bandwidth for other applications and (2) it reduces the degree of higher level retransmissions. When the input rate is reduced for the congested VCs, traffic sharing non-congested links with the congested VCs can take advantage of the saved bandwidth in these non-congested links. These applications can put more data into the network and the overall performance of the network is then increased. Our preliminary result shows that, with the configuration discussed in section 5.6, an average of thirteen percent improvement is achieved by the SPD scheme. Secondly, when the input to the congested links is reduced, the degree of retransmission is also reduced.

1.3 Overview

Chapter 2 of this document first discusses some of the relevant flow control mechanism for TCP and ATM protocols. Methods for TCP's adaptive retransmission schemes such as: retransmission timer and backoff mechanism, window-based flow control, congestion avoidance and control, etc. are presented. The second part of Chapter 2 covers the five

1. Goodput is the effective throughput which constitutes the portion of input that is successfully transmitted by the source and is also successfully received by the receiver.

traffic management and different service categories such as: the constant bit rate service, the variable bit rate service, the unspecified bit rate services, and the available bit rate services. The third part of Chapter 2 discusses the internetworking issues of TCP over ATM networks, especially on the ABR and UBR services. The last part of Chapter 2 presents previous research in the area of TCP over ATM networks which is followed by a discussion on the existing EPD schemes.

Chapter 3 introduces the proposed single packet discard scheme (SPD), the motivation, the key idea, and the implementation details. The pseudo-code of the SPD implemented for the UBR switches can be found in section 3.3. Chapter 4 covers the simulation setup. The simulation tool used in this thesis is the Network Simulator (Netsim). This chapter discusses important components used, such as the persistent and bursty source components, the TCP components, the physical link components, the ATM edge device, and the ATM switches. Important parameters for these components are also covered and is followed by the definition of the performance evaluation parameter "goodput." Chapter 4 ends with a brief discussion on the four goals for the simulation works in this thesis.

Chapter 5 presents the results and discussion of the simulations. The first set of simulation explores where the most optimal position is for the dropping threshold of SPD. The second set compares the goodput to input ratio of SPD and EPD schemes for the persistent sources. It is followed by another set of simulations focused on bursty data sources in section 5.4. Section 5.5 shows the switch queue for EPD and SPD schemes which gives insights to how SPD really works. Lastly, a set of simulations is presented to show that overall goodput increases from EPD to SPD scheme. Chapter 6 ends with concluding remarks regarding the research and future works.

Chapter 2

Flow Control Mechanisms: an Overview

2.1 TCP Flow Control -- Adaptive Retransmission Mechanisms

Typical TCP data flow is characterized by bursts, provides window-based flow control, and uses adaptive retransmission to accommodate the diversity of the underlying physical network. Adaptive retransmission basically predicts the future behavior of the traffic based on its recent traffic pattern. Five mechanisms are used by TCP to accomplish adaptive transmission: (1) retransmission timer and backoff, (2) window-based flow control, (3) maximum segment size computation, (4) congestion avoidance and control, and (5) round-trip delay estimation[2]. These sections briefly describe each of the mechanisms.

2.1.1 Retransmission Timer and Backoff Mechanism

In the retransmission timer and backoff mechanism, the source TCP assigns a sequence number to every packet that it sends to the receiver TCP. Once the receiver successfully receives the packet, it sends an acknowledgment to the sender with an acknowledgment number. The acknowledgment number has the same value as the sequence number of the sent packet. Upon receiving the acknowledgment, the sender knows that this particular packet has been successfully transmitted and received.

Prior to sending any packet, the sender sets a retransmission timer, and if no acknowledgment is received after the timer expires, the sender retransmits this particular packet. TCP assumes that if no acknowledgment packet is received, then the packet is probably lost. Once a lost packet is detected, TCP reduces the retransmission and allows tolerance of a longer delay by doubling the retransmission timer using the Karn's algorithm[3].

2.1.2 Window-based Flow Control

Along with the acknowledgment number, the value of advertisement window is also sent back to the source TCP. The advertisement window tells the sender how much buffer space the receiver has available for additional data. The sender uses this number to modify its congestion window which constrains how many packets the sender can send beyond the previously acknowledged packet. During congestion, the advertisement window is small, and fewer packets are sent to the network. This method is then used to control the flow of data across a connection.

2.1.3 Maximum Segment Size Computation

During the TCP connection establishment (three-way handshake), a maximum segment size (MSS) is agreed upon between the sender and receiver of the TCP connection. This MSS number does not change over the course of the connection, thus the receiver knows the size of the largest possible segment that can be expected to arrive.

2.1.4 Congestion Avoidance and Control

During congestion, transmission delays increase and cause retransmission of missing segments. Without any congestion control, the retransmission mechanism could go into positive feedback, worsen the delay, and lead to congestion collapse. TCP uses "multiplicative decrease" to avoid adding more data into the network when packet or data loss occurs. "Multiplicative decrease" basically reduces the sender's congestion window size by a multiple constant. This alleviates the number of packets that a sender puts into the network during congestion.

If TCP is coupled with an underlying transport layer which has its own flow control mechanism, such as in TCP over ATM, one way to trigger TCP slow-down is to delay the receiver's acknowledgment by either holding the acknowledgment at the transport level or purposely dropping a packet which causes the TCP timer to expire while waiting for the

acknowledgment. This is the underlying idea of this thesis and will be discussed in more detail in later sections.

TCP also uses "slow-start" and "slow-increase" to limit congestion during the start and recovery stage of the connection. Slow-start is the reverse of multiplicative decrease. When it first starts, there is one segment in the congestion window. If the transmission is successful, the number of segments in the congestion window exponentially increases until a threshold is reached. Once the threshold is reached, slow-start is ended and slow-increase starts. During the slow-increase phase, one segment is added to the window per roundtrip time. The slow-start enables better utilization of the bandwidth and the slow-increase allows the avoidance of congestion.

2.1.5 Round-trip Delay Estimation

The performance of TCP's adaptive retransmission depends on how accurately it estimates the round-trip delay. According to Karn's algorithm, the round-trip delays contributed by the retransmission do not count towards the estimation. The TCP uses a fast mean update algorithm to keep a "running average" of the round-trip delay with higher weights on more recent measurements.

2.2 The ATM Traffic Management and the Five Services Categories

ATM traffic management specifies five service categories [3]: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non-real-time variable bit rate, unspecified bit rate(UBR), and the available bit rate service (ABR). Each service gives a different specification of the Quality of Service (QoS) for different traffic characteristics. A brief description on each traffic type is followed.

2.2.1 The Constant Bit Rate Service

The constant bit rate service (CBR) is used by any application which requires the availability of a continuously constant amount of bandwidth during the entire duration of

the connection. CBR is then intended for real-time applications, such as voice and video applications which require tightly constrained delay and delay variations. For those applications, cells which are delayed beyond the value specified by the cell transfer delay (CTD) have significantly less value to the application.

2.2.2 The Real-Time Variable Bit Rate Service

The real-time variable bit rate service (rt-VBR) is used by real-time applications that also require tightly constrained delay and delay variation, such as voice and video applications. Unlike the applications using CBR service, applications using rt-VBR service expected to transmit at a rate varying with time, such that the application can be bursty. Similar to CBR applications, rt-VBR cells delayed beyond the CTD have little significance. The rt-VBR service also supports statistical multiplexing of real-time sources and provide a consistently guaranteed QoS.

2.2.3 Non-Realtime Variable Bit Rate Service

Non-real-time variable bit rate service (nrt-VBR) is used by non-real time applications which have defined bursty traffic characteristics. The bandwidth requirement for nrt-VBR fluctuate like the nt-VBR service, but nrt-VBR is more suitable for other applications that do not require the real-time delivery.

2.2.4 Unspecified Bit Rate Service

The unspecified bit rate service (UBR) is used for non-real-time applications that can tolerate both delay and delay variations. The most notable applications using UBR are traditional computer communication applications, such as file transfer and electronic-mail. UBR sources are expected to transmit non-continuous bursts of cells. The UBR service does not specify traffic related to the service guarantees. UBR service is considered as the "best effort service" in the cell level; common congestion control for UBR may be performed at a higher layer than on an end-to-end basis.

2.2.5 Available Bit Rate Service

Similar to the UBR service, the available bit rate service (ABR) is used by communication applications which have the ability to reduce or increase their information transfer rate depending on the bandwidth availability. There are no deterministic traffic parameters for those applications, but users should be willing to live with unreserved bandwidth. In contrast to the UBR service, the ABR service has a specified rate of flow control. This flow control supports several types of feedback to control the ATM network source rates in response to the current ATM network traffic characteristics. If an end-system adapts this feedback control, the end-to-end traffic is expected to have low cell-loss rate and a fair share of the available bandwidth.

2.3 Internetworking of TCP over ATM

With the ever increasing practical importance of running TCP over ATM network, much work has been done both in understanding the interaction between TCP and ATM, and in finding ways to improve its end-to-end performance. The first part of this chapter describes some of the important concepts in the area of TCP over ATM internetworking. The second part describes some research on creating mechanisms to improve end-to-end performance on which this thesis is based.

Although much research done on TCP over ATM has assumed an end-to-end ATM network environment, most data applications are currently supported over legacy networks (e.g. Ethernet) interconnected with ATM networks (e.g. ATM backbone). Although ATM is designed to handle integrated services including both traditional data services and real-time multimedia applications, TCP/IP is primarily designed for data applications such as electronic-mail and file transfer. On the contrary, the most of the CBR/VBR services are designed for real-time and QoS intensive applications which most likely will rely on other types of layer 3 protocols rather than TCP/IP. Thus, the focus of this section is on intercon-

necting TCP and ATM networks using ABR or UBR service. The following section talks about the pros and cons in using these two services.

2.3.1 ABR versus UBR in Flow Control

As described in Chapter 1, ABR employs a rate-based feedback flow control algorithm to avoid severe network congestion. This minimizes the buffer requirement and cell loss rate at the individual switches within the ATM network. However, it is conceivable that ABR simply pushes the congestion from the ATM switches to the ATM edge devices. Thus, many people argue that the reduction in the ABR switch buffer requirement is at the expense of an increase in the buffer size required for the ABR edge device.

On the other hand, UBR employs no flow control mechanisms to deal with the congestion. When data is available for transmission at the source, it can be sent as fast as the possible link speed. The transmission rate into the ATM network is not constrained by the state of congestion. However, if the buffer overflows out of the ATM switch buffer, cell losses may occur. Consequently, UBR depends on higher layer such as TCP to handle flow control and recovery of data (i.e. retransmission of TCP packets). Thus, UBR is much easier to implement both at the ATM switches or at the ATM edge devices since it does not require an extra scheme to pass the congestion information from one end of ATM network to the other. Thus, many people argue that UBR is equally effective and much less complex than ABR service.

2.3.2 TCP/IP over UBR Service

The top part of Figure 1 shows an example of the packet flow in an UBR-based ATM subnetwork interconnected with a TCP network. The lower part of this figure shows that the TCP flow control is extended from end to end. In this case, there is no internal ATM flow control mechanism to interfere with the end-to-end TCP flow control mechanisms. When congestion occurs in the ATM network cloud, cells can be dropped which causes

the reduction of the TCP congestion window. Once cells are dropped, TCP is responsible for retransmitting the corresponding packet from the source side to the destination side.

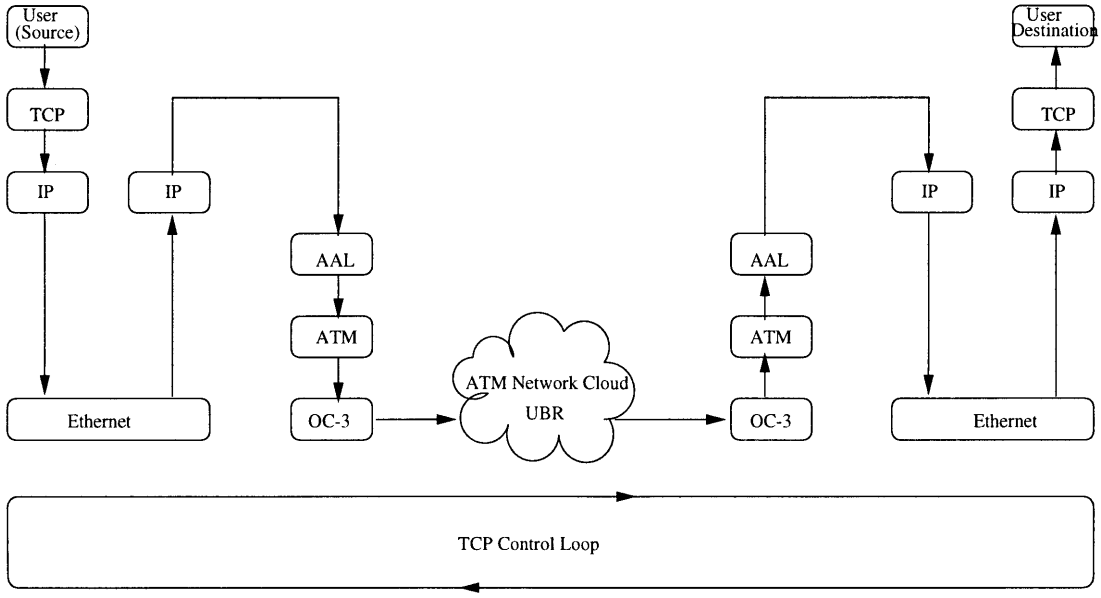

Figure 2.1: TCP Interconnects with UBR Service

2.3.3 TCP over ABR Service

Figure 2 shows the packet flow in an ABR based ATM subnetwork within a TCP network. In this case, when congestion occurs in the ATM network, ABR rate control ensures that the ATM edge device reduces its transmission rate into the ATM network. If congestion continues, the buffer in the edge device becomes full and starts dropping packets. This signals to the source TCP flow control that the network is congested. There are two flow control loops involved with TCP over ABR service, and this causes a longer delay passing the congestion information from ABR control to TCP control. In terms of getting TCP advanced notification of the state of congestion, ABR control actually does not perform as well as UBR service, because it could prolong the delay for notification.

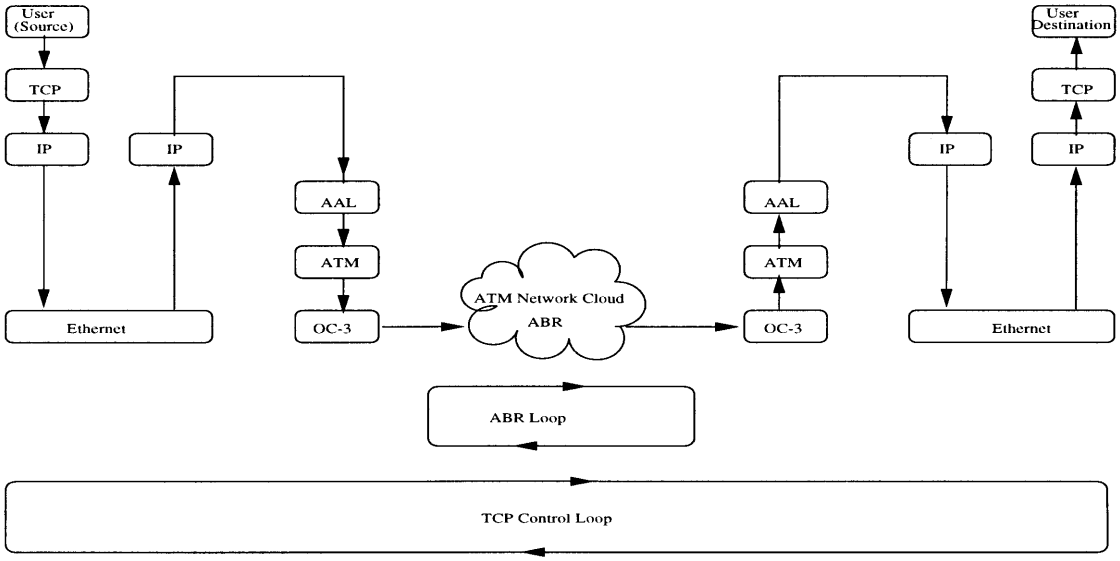

Figure 2.2: TCP interconnects with ABR service

2.4 Previous Research

Yin and Jagannath [2] studied the performance of TCP over both binary both mode ABR (with EFCI Switches) and UBR in an IP/ATM internetworking environment. Ren, Siu, et al. [4] extended their work and used simulations to evaluate the performance of TCP over ABR (explicit rate and binary mode) and UBR (with and without EPD schemes) Narvaez and Siu [5] proposed a scheme which matches the TCP source rate to the ABR explicit rate by controlling the flow of TCP acknowledgments at the network interface. In this section, two researches laying the foundations for this thesis are discussed.

By analyzing their simulation results, Ren, Siu, et al. [4] provide a comprehensive study and evaluation for the performance of TCP in four different schemes: ABR explicit-rate (ER), ABR binary-mode, UBR, and UBR with EPD. From understanding their work, it can be learned that, in general, the ABR service requires larger buffer size in the edge device because ABR tends to push congestion to the edge of the ATM subnetwork. On the other hand, the UBR service requires larger buffer sizes in the individual ATM switch to

have similar end-to-end performance to ABR service. When there are no cells being dropped at the switches, UBR guarantees all virtual circuits (VCs) sending cells at the peak rate. Thus, UBR is less likely than ABR to cause buffer overflow at the edge devices due to UBR's fast draining of data. However, when there is buffer overflow in the switch, some bandwidth is wasted. EPD is a common technique for improving the goodput of packet based traffic (i.e. TCP) over the UBR service. However, for transmission with a long duration, congestion could occur when the TCP window increases to a very large size. While congestion forces the switch buffer to exceed the EPD threshold, the ABR service drops all the incoming packets until the buffer is reduced to at least a packet less than the EPD threshold. During congestion, the rate for cells leaving the buffer is less than that for new cells entering to the buffer. Thus, the ATM switch is often forced to drop cells that constitute at least one entire TCP window. Should this situation occurs, the end-to-end TCP goodput is largely reduced.

In the same paper, a TCP acknowledgment holding mechanism is introduced. In this scheme, the ATM edge device temporarily holds the TCP acknowledgment packet to reduce the TCP window if the edge device buffer rises above a preset threshold. Among the two variations of the scheme (dynamic-time holding and fixed-time holding) the fixed-time holding is more responsible for controlling the edge queue.

Another set of researchers attempt to pass the ATM level congestion status to TCP from delaying the acknowledgment of packets. Narvaez and Siu [5] calculated the explicit rate required in the ABR service and use that information to control the rate for backwards TCP acknowledgment. If there is no congestion, ABR's explicit rate is high and the edge device sends the acknowledgment back to the source immediately. TCP then increases its window size as usual. However, if congestion does occur, ABR's explicit rate is low and the edge device holds the backwards acknowledgment much longer. TCP then

slows down its rate, and hopefully alleviates the congestion. This ABR scheme is more difficult to implement than the one described earlier.

Both research works have a common goal, which is to pass on the congestion indication to TCP's flow control. This is also the basis of this thesis. The difference is that in this thesis, no mechanisms were used to delay the acknowledgment cells. Since delay mechanisms also takes up some network resources, this thesis presents the idea of dropping one packet at an earlier stage of congestion to trigger the TCP's flow control to slow down its input rate in to the ATM network, thereby which increasing the goodput to input ratio and further increasing the efficiency in using the network resources.

2.5 Existing Early Packet Discard Schemes

TCP's end-to-end congestion control is activated when a packet is lost in the network. It alleviates congestion by reducing the TCP congestion window. For the UBR service without the EPD scheme, TCP's end-to-end congestion control is only activated when the buffers overflow after a long period of congestion. The key idea of EPD is to preset a threshold at the ATM switches. If there is a possibility that the switch buffer might overflow when a whole packet of cells are admitted to the buffer, the switch drops the entire packet prior to preventing the buffer overflow. This also prevents the congested link from transmitting useless cells and reduces the total number of corrupted packets. EPD also allows TCP to react to congestion within a shorter delay than the simple UBR service without EPD.

There are two versions of EPD. The first one is called aggregate EPD, or simple EPD, and the second one is called EPD with per-VC accounting. For aggregate EPD, the pre-set threshold is placed in the output buffer of the ATM switch. For example, the EPD threshold could be 700 cells in a buffer of size 1000 cells. If the buffer occupancy reaches over 700 cells, the switch drops the first arriving cell and all subsequent cells of the incoming

packet. As long as the buffer occupancy exceeds the 700-cell threshold, the switch continues to drop packets from any connections that have incoming packets.

The EPD with per-VC accounting keeps track of the individual buffer occupancy of each connecting VC. Another per-VC threshold is set, usually to the aggregate threshold divided by the number of active VC. Continuing with the previous example, there are ten active VC's in the network. The per-VC threshold is set to 70 cells. For this scheme, packets are dropped only under two conditions: The first condition is that the aggregate buffer size exceed the aggregate threshold. The second condition is that the per-VC queue occupancy exceed the per-VC threshold. The second version is to ensure fairness in that the EPD drops packets on a rarely active VC because other active VCs are making the network congested.

In the packet-switched network, it is easy to discard a whole packet. However, in the cell-based ATM network, only cell-dropping can occur. While implementing EPD in the cell level, an end of message (EOM) flag in each cell is used to distinguish whether a cell is the last one of an AAL5 frame (last cell of the packet). This could then be used to distinguish between cells of one TCP packet and cells made of adjacent TCP packets. This way, it is possible to identify the group of ATM cells that are belonging to one TCP packet and dropped cells making up the same packet if necessary.

Chapter 3

The Proposed Single Packet Discard Scheme

3.1 Motivation

As stated earlier, this thesis aims to design a mechanism for improving the end-to-end performance of IP over ATM interconnected networks. In an interconnected network, resources transmitted from the source first pass through the TCP packet network are transferred via the ATM subnetwork before they reach the destination TCP packet network. Any cells that are dropped in the ATM switch generate wasted resources on the TCP network and the travelled ATM network.

When the ATM network is congested as the TCP window enlarges, EPD drops all of the incoming packets in the TCP window until TCP slows down. It takes a round-trip delay before TCP can react to the first dropped packet by decreasing the rate, so that during this time, all packets in the TCP window are likely to be dropped. These large amount of packets, which are now being converted to AAL5 frames, have already been used by the TCP and ATM resources. The motivation of single packet discard scheme (SPD) is to avoid these large cell drops by the EPD scheme.

3.2 Description of the Single Packet Discard Scheme

The key idea of single packet discard (SPD) is to drop a small amount of cells, either one packet in length or less, at an early stage of congestion. This gives TCP advanced notification about the state of congestion of the ATM subnetwork. Once this notification reaches TCP, the TCP window decreases either to half of the previous window size or to one packet size depending on the TCP version. If the notification reaches the TCP early enough, the buffer will never overflow because the TCP input to the ATM network is reduced once the size of the TCP window is reduced. Comparing the small number of cells

being dropped using SPD to the large number of cells dropped by SDP of cells, it is obvious that the TCP badput is minimized to the purposely dropped cells. Both the goodput and the goodput to input ratio of the ATM subnetwork which are important measures for the TCP end-to-end performance.

The reason to drop a single packet is the same as that explained for EPD and PPD. It is wasteful to discard one cell and not to discard those cells belonging to the same packet. Another aspect of the scheme is to drop the cell only once and will not drop again until the buffer is empty again. Since it takes some delay before TCP times-out and slows down the rate, the buffer is still being built up after the dropping of cells. During this time, there is no need to drop another set of cells since it has no need to signal TCP to decrease the rate. Thus, the scheme is designed to minimize the extra dropping, which means no dropping of cells until the buffer is empty again.

From looking at the switch buffer occupancy, one can roughly tell whether the connected link is congested or not. If the buffer is empty, there is no congestion. At the beginning of congestion, the buffer occupation builds up. With the SPD scheme, one incoming packet is dropped if the buffer occupation is greater than the pre-set threshold, resulting in the SPD "dropping threshold." While that one packet is dropped, the following incoming packets are continuously accepted to the buffer. At this time the buffer occupation for the connection continuously builds up. Until TCP times out, which causes the TCP rate to slow down, the buffer starts to decrease (although with possible fluctuation) to zero and congestion is then alleviated. After the next time the buffer builds up, the SPD mechanism will drop one more packet and the whole process is repeated again.

During the time after the one packet is dropped and before TCP times out, the buffer fills up. To avoid buffer overflow, which is the worst situation, the EPD scheme is included since it has a different purpose from SPD and can be used concurrently. As stated earlier,

careful attention needs to be paid for dropping the group of cells belonging to the same packet. AAL5 does not support special information in identifying the first cell of the packet, but with this implementation, it is possible to indicate when the current cell is the first cell.

3.3 The Implementation of single packet discard Mechanism

Our ATM switches employ the following pseudo-code to simulate the SPD mechanism. This code is placed in the ABR switches and is executed whenever a cell comes in to the switch.

```
   /* Initialization: */
   Receiving = FALSE;
   Discarding = FALSE;
   Dropping = TRUE;

   /* When a cell is coming to an ATM switch: */
   if (q == 0) /* The Active VC has not built up anything in the buffer
*/
         Dropping = TRUE;

   if the incoming cell is NOT the last cell
         if (Receiving == FALSE) /* The incoming cell is the first cell
of a packet*/
               if (Q < Threshold)
                     if (q < SPDth)
                           accept the cell;
                           Receiving = TRUE;
                     else /* (q > SPDth) */
                           if ((Dropping == TRUE) && (Q > CongestTh))
                                 discard the cell;
                                 Discarding = TRUE;
                                 Receiving = TRUE;
                                 Dropping == FALSE;
                           else /* Dropping == FALSE or Q <= CongestTh*/
                                 accept the cell
                                 Receiving = TRUE;
               else /* Q > = Threshold */
                     discard the cell;
                     Receiving = TRUE;
                     Discarding = TRUE;
```

```
        else /* Receiving == TRUE */
                if (Discarding = TRUE)
                        discard the cell;
                else /* Discarding == FALSE*/
                        if (Q >= Qmax)
                                discard the cell;
                                discarding = TRUE;
                        else /* Q < Qmax */
                                accept the cell;
else /* the last cell */
        Receiving = FALSE;
        Discarding = FALSE;
        if (Q < Qmax)
                accept the cell;
        else /* Q >= Qmax */
                discard the cell;
```

In the pseudo-code, the variable Q represents the current queue size of the entire output buffer. The variable *Qmax* is the buffer size of the switch. The variable q represents the current buffer occupancy of the VC to which the current incoming cell belongs. The variables *Receiving*, *Discarding*, and *Dropping* are per-VC states.

The variable *Receiving* indicates whether the buffer is receiving data. If *Receiving* equals to TRUE, then this AAL5 frame is being received. If *Receiving* equals to FALSE, then this AAL5 frame has not yet being received. The last cell can change *Receiving* from TRUE to FALSE, because after accepting the last cell, it has not yet determined whether to accept the next frame. When a cell arrives and *Receiving* is equal to FALSE, then this incoming cell is the first cell of the packet. If SPD decides to accept this cell, it sets *Receiving* to TRUE. When a cell comes in and *Receiving* is equal to TRUE, then SPD knows that this cell is one of the body cells or EOM cells.

The variable *Discarding* indicates whether the buffer is discarding. If *Discarding* equals to TRUE, this AAL frame is being discarded. If *Discarding* equals to FALSE, then the current AAL frame is not being discarded.

The third state variable *Dropping* indicates whether single dropping the packet is an option (*Dropping* = TRUE) in the immediate future. Once a packet is being dropped, the ATM switch waits until the buffer is empty, thereby it allowing the single-dropping option again. Hence, at the beginning of the code, it checks whether q is equal to zero. If q equals to zero, it means that the single drop option can be opened. As another condition, SPD is only activated if the current output buffer size is greater than a minimum SPD activation threshold. SPD waits until the entire queue buffer is somewhat congested before single dropping the packet.

Chapter 4

Simulation Setup

4.1 Network Simulator (NetSim)

The simulation tool used in this thesis is called the Network Simulator (NetSim)[1]. NetSim is used to simulate the SPD scheme and the results obtained from NetSim were then used to observe the performance of the SPD scheme.

NetSim is an event-driven simulator that is composed of various components which send messages to one another. The important network components used in this thesis include the following: (1) an user component, (2) a TCP component, (3) a multiplexing and de-multiplexing component, (4) an ATM edge component, (5) a SONET OC-3 link component, (6) and ATM switch component. The ATM edge component performs ATM Adaptation Layer for data services including segmentation and reassembly (SAR) of TCP packets. The ATM switch used in this thesis models an UBR switch combined with EPD and SPD capability. Each component schedules events for neighboring components which is necessary for data transmission. The action of the component is entirely determined by the code controlling the components. The interaction of each component models the interaction between the real network hardware components by the protocol written by the standard committees. The simulator also provides the means to display the topology of the network and parameters of its operation.

Figure 4.1 shows the network setup of the simulation. In the figure, there are six peer-to-peer connections with traffic going from six source users to the six destination users. In the different sets of the simulations, the number of peer-to-peer connection is different, but the overall layout of network architecture is the same.

1. NetSim is the before-commercialized version of OPNET, a very popular commercial network simulator.

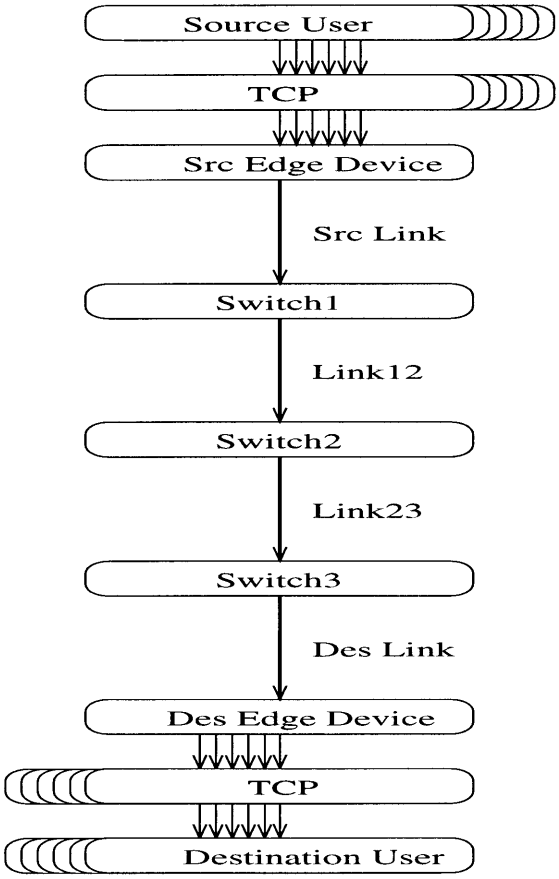

Figure 4.1: Simulation Network Architecture Setup

For each connection, the user component generates either persistent or bursty data. When TCP is ready for more data and the user also have available data, TCP converts these data into TCP packets and passes them onto the ATM edge devices for AAL5 processing. The ATM switches then perform cell switching between their input and output ports. On the receiving side, cells are reassembled and passed onto the corresponding TCP and user component. Each component and its characteristic parameters are described briefly in the following sections.

4.1.1 User Component

The user component can either be a persistent or a bursty source. A persistent source always has data to send, and it sends out as fast as TCP allows. Most of the simulations use the persistent source to model the extreme traffic load. Bursty source is also used to verify that similar results can be carried to a more realistic environment.

A bursty source sends out data according to a Poisson probability distribution function. In our simulation, a random generator is used to generate uniformly distributed data. The inter-arrival time for the user to generate data can be calculated as followed. The PDF of the exponentially distributed inter-arrival time is $P(t) = \mu e^{-\mu t}$, where t is the inter-arrival time and μ is the service rate (1/mean). The PDF of the random generator is $P(r) = \frac{1}{Rm}$,where Rm is the maximum random value and r is the generated random number divided by Rm. To find the mapping function $f(r)$ from r to t, we set the CDF of the two probabilities to equal.

$$\int_{r1}^{r2} \frac{d}{dr} P(r)dr = \int_{f(r1)}^{f(r2)} P(t)dt$$

$$\int_{r1}^{r2} 1 dr = \int_{f(r1)}^{f(r2)} \mu e^{-\mu t} dt$$

Let $r1 = 0$,

$$r = \int_{-\infty}^{f(r)} (-\mu e^{-\mu t})dt$$

$$r = -\mu e^{-\mu f(r)} - (-\mu e^{(-\mu \cdot -\infty)})$$

$$r = e^{-\mu f(r)}$$

$$\ln r = -\mu f(r)$$

$$f(r) = \frac{1}{-\mu} \ln r$$

The bursty source user then sends out the next data based on μ, the result from the random generator, and the function *f(r)*.

In the simulation, the transmission size is chosen as 1024 Kbytes for both types of the users. The unit for μ is 1024 bytes per micro-second. If μ is set to one, the user can persistently produce 1024 Mbytes of data every one second. For the bursty source, μ is set to be less than one. The smaller the μ, the smaller the average rate for the bursty source.

4.1.2 The TCP Components

This thesis uses the Tahoe version of TCP. In this version, the TCP window reduces to one packet size during the recovery stage. The three important TCP parameters are mean processing delay, the packet size, and the window size. This thesis uses the typical values:

Mean Processing Delay = 300 μ sec
Packet Size = 1024 Bytes
Max Window Size = 64KBytes

The Max Window Size is the maximum TCP window size and during the actual simulations, the window size never reaches its maximum possible value.

4.1.3 The Physical Links

There are two types of links in the simulation. The first type is the source or the destination link. These links connect between the ATM edge devices and the rest of the ATM network. The speed of the link is set to 155 megabits per second and the delay is set to 50 nanoseconds. The other type of link is the link in-between the three ATM switches. Link12 is between Switch1 and Switch2. Link23 is between Switch2 and Switch3. The

delays for the links are 3 milli-seconds. The longer delays, as compared to the source/destination links, model longer physical distances between these switches. The speed for Link12 is 155 megabits per second. The speed for Link23 is 10 megabits per second. Link23 is set to a slower speed to ensure the congestion state.

4.1.4 The ATM Edge Device

With the code-name, Host, the ATM Edge Devices component have SAR capabilities. Each packet has a segmentation and re-assemblage delay of 300 nanoseconds. The edge device is where ABR service implementation occurs. Since we are only using UBR service, the ABR service mechanisms were not used.

4.1.5 The ATM Switches

The are three switches in the simulation. Each of them are labeled with 1/2/3 according to Figure 4.1. Each switch is identical with the following parameters.

Output Buffer Size = 1000 cells
EPD Threshold = K* Output Buffer Size
single packet discard Threshold = 200 cells
single packet discard at q = 20 cells.

K is the EPD constant. If K equals to 0.5, then EPD threshold is set to be 500 cells.

4.2 An Explanation on Goodput

The primary measure in this thesis is the ratio of goodput versus input. When ATM transports the cells from the higher-level protocols such as TCP, it is important to consider the impact of ATM cell loss on the effective throughput of the higher-level protocols. Even if one particular cell is successfully transported to the other end of the ATM network, if the other cells belonging the same frame were dropped, that particular cell will ultimately be discarded, thereby wasting the bandwidth used to transport that particular cell. The traffic corresponding to these discarded cells are termed as the "badput." "Badput" is the opposite of to "goodput", which refers to cells that constitute packets that are successfully

transported to the destination by the higher level protocol.

The "goodput" in our configuration is calculated to be the number of cells received by the receiving switch (switch 3), lessened by the discarded cells, which constitute some part of badput. The "input" in our configuration is determined to be the number of cells delivered by the sending switch (switch 1). One goal is to improve the ratio of goodput to input. Another is to lessen the congestion state while achieving a high goodput and input ratio because a high goodput and input ratio gives us high bandwidth efficiency. The bandwidth that is not used can always be allocated for other applications.

4.3 Four Goals

The simulations in this thesis aim to achieve four different goals. The first goal is to find the position where single dropping is optimal. The second goal is to verify that single dropping gives a better end-to-end goodput to input ratio than that of both aggregate EPD and aggregate plus per-VC EPD for persistent sources. The third goal is to verify that single dropping also gives better end-to-end performance for bursty sources since bursty source resembles more of the 'realistic' network. The last goal is to show that OSD improves not only the goodput to input ratio but also the overall goodput in a network with traffic of different paths.

Chapter 5

Simulation Results and Analysis

5.1 Overview

Four sets of simulation results are presented and discussed in the following five sections. Section 5.2 discusses the relationship between the SPD threshold and the goodput to input performance of the scheme. Section 5.3 presents a serious of goodput to input comparisons between the two EPD schemes and the SPD schemes for persistent traffic. Section 5.4 presents results of similar simulations done for bursty sources. Section 5.5 shows how the buffer size of the congestion switches fluctuates and how it varies differently for the different packet discarding schemes. Lastly, in section 5.6, a new topology is designed to show that SPD improves the sum end-to-end goodput.

5.2 The Dropping Threshold of SPD

The first set of simulations explores finding the most optimal position for the "dropping threshold" of the single packet discard scheme. After the queue occupancy for a VC reaches beyond the SPD dropping threshold, the UBR switch drops the one incoming packet coming from that VC. The simulations help to define the relationship between the goodput to input ratio and the SPD thresholds. There are two speculations for the optimal positions. The first stipulates that the optimal position is some fixed proportion of the total buffer size for the switch divided by the number of VCs, which we calls the "VC allocation size." The second speculates that the optimal position is independent of the "VC allocation size."[1] Also, since these simulations is determine the major characteristic of the SPD scheme, by decreasing the degree of interaction between SPD and EPD reduce the

1. Although the "VC allocation size" is defined as the total switch buffer size divided by the number of VC, it does not mean that each VC can only occupy up to the "VC allocation size." This variable is merely defined for discussion purposes.

EPD interference. Therefore, the results can be attributed solely to SPD. The EPD threshold factor K is set to be one in order to deactivate the EPD scheme in this set of simulations.

There are six configurations for this set of simulations. Each with a different number of VCs. They are 15, 20, 25, 30, 35, and 40 peer-to-peer VCs. The rest of the parameters, as described in Chapter 4, are kept the same for all configurations. For example, the switch queue is kept constant to 1000 cells. With different number of VCs, each configuration has different "VC allocation size." For example, for 15VCs, the "VC allocation size" is 1000 divided by 15, 67cells.

Each SPD dropping threshold is set to some fractions of the VC allocation size. For example, for 15 VCs, the threshold is set for each VC to be 0.2, 0.4, 0.6, and 0.8 of 67 cells. The exact position is not as important as the trend of the end-to-end goodput to input ratio performance. The results show that if the number of VCs is large, the performance increases as the fractional setting decreases. In contrast, if the number of VCs is small, the performance increases as the fractional setting increases. After plotting the values, it is shown that all the optimal position are around 20 cells regardless of the number of VCs in the simulations.

To verify the results, the same set of simulations were then run again with the dropping threshold set to 18, 19, 20, and 21 cells. It is found again that the end-to-end performance is highest at 20 cells. Also, similar results can be seen for a dropping threshold set to 19 cells. All the simulation results are plotted in Figures 5.1 - 5.6. The exact values are also tabulated in the following figures.

The plots show the goodput to input ratio versus the SPD dropping threshold in number of cells. Each data represents one simulation result. Referring to figure 4.1, The goodput is measured as the number of cells sent from Switch2 to Switch3 that belongs to a

complete packets. The input is measured as the number of cells sent from switch 1 to switch 2.

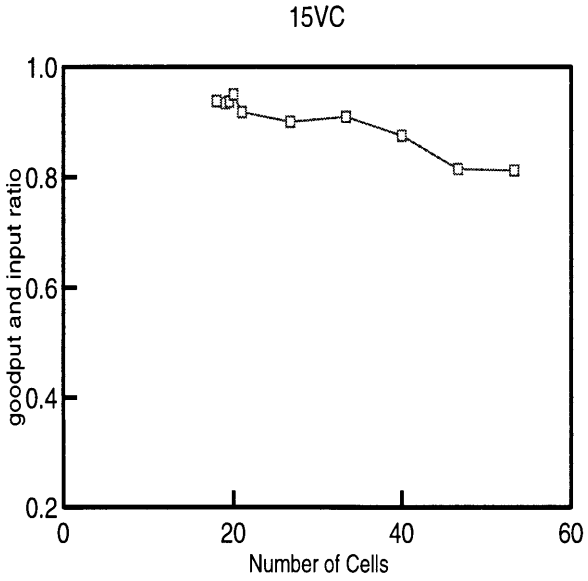

Figure 5.1: SPD Goodput to Input Ratio versus Dropping Threshold for 15VCs

Position for Single Packet Dropping (cells)	Goodput to input ratio
18	0.938
19	0.934
20	0.950
21	0.918
27	0.9003
34	0.9095
40	0.8752
47	0.8148

Table 5.1: SPD Goodput to Input Ratio versus Dropping Threshold for 15VCs

Position for Single Packet Dropping (cells)	Goodput to input ratio
54	0.812

Table 5.1: SPD Goodput to Input Ratio versus Dropping Threshold for 15VCs

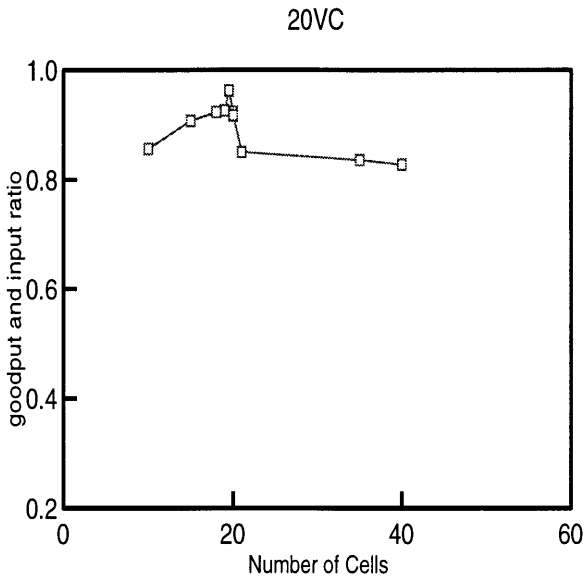

Figure 5.2: SPD Goodput to Input Ratio versus Dropping Threshold for 20VCs

Position for Single Packet Dropping (cells)	Goodput to Input Ratio
10	0.8566
15	0.9081
18	0.924

Table 5.2: SPD Goodput to Input Ratio versus Dropping Threshold for 20 VCs

Position for Single Packet Dropping (cells)	Goodput to Input Ratio
19	0.927
20	0.9243
20	0.918
21	0.851
35	0.8363
40	0.828

Table 5.2: SPD Goodput to Input Ratio versus Dropping Threshold for 20 VCs

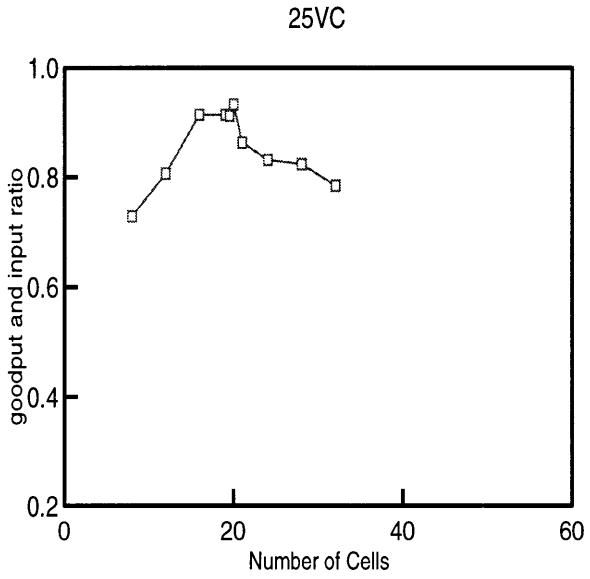

Figure 5.3: SPD Goodput to Input Ratio versus Dropping Threshold for 25 VCs

Position of Single Packet Dropping (cells)	Goodput to Input Ratio
8	0.7284
12	0.8063
16	0.914
19	0.914
20	0.9327
21	0.863
24	0.8316
28	0.8236
32	0.7843

Table 5.3: SPD Goodput to Input Ratio versus Dropping Threshold for 25 VCs

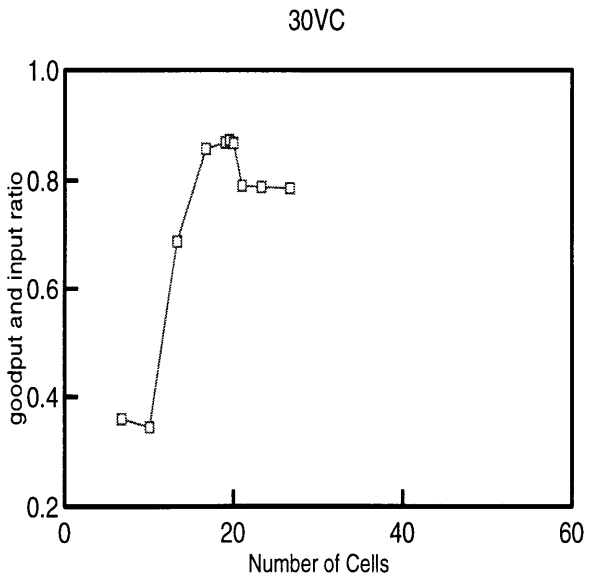

Figure 5.4: SPD Goodput to Input Ratio versus Dropping Threshold for 30VCs

Position of the Single Packet Dropping (cells)	Goodput to Input Ratio
7	0.3589
10	0.344
14	0.6889
17	0.8576
19	0.869
20	0.868
21	0.791
24	0.7885
27	0.7863

Table 5.4: SPD Goodput to Input Ratio versus Dropping Threshold for 30VCs

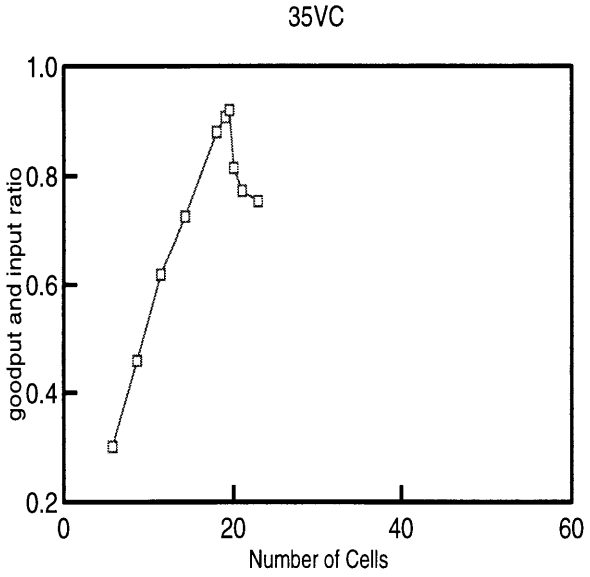

Figure 5.5: SPD Goodput to Input Ratio versus Dropping Threshold for 35 VCs

The Position of Single Packet Dropping (cells)	Goodput to Input Ratio
6	0.3006
9	0.4592
12	0.6186
15	0.7259
18	0.880
19	0.907
20	0.815
21	0.773
23	0.7541

Table 5.5: SPD Goodput to Input Ratio versus Dropping Threshold for 35VCs

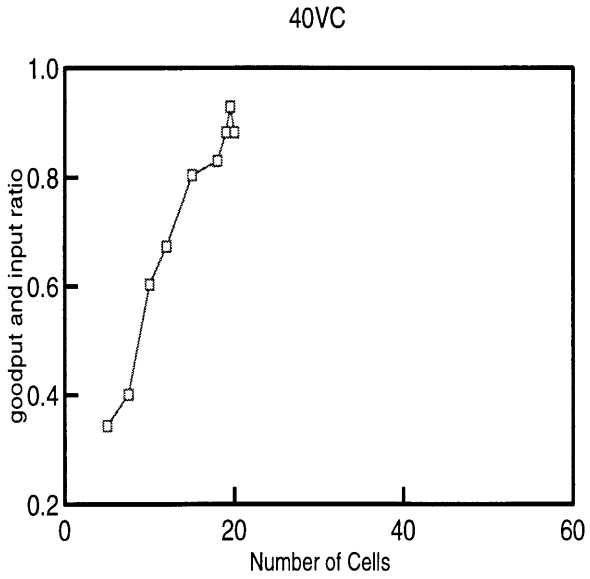

Figure 5.6: SPD Goodput to Input Ratio versus Dropping Threshold for 40 VCs

The Position for Single Packet Dropping (cells)	Goodput to Input Ratio
5	0.3428
8	0.4007
10	0.6033
12	0.6732
15	0.804
18	0.830
19	0.882
20	0.882

Table 5.6: SPD Goodput to Input Ratio versus Dropping Threshold for 40 VCs

The results show that the optimal SPD dropping threshold is around 20 cells for the simulation configuration described in Chapter 3. With 1024 bytes as the TCP buffer size and 53 bytes as the ATM cell size, 20-cell is approximately the length of 1 packet (1024/ 53 = 19.3). The simulation results also show that if the dropping position is set below 20 cells, the goodput to input ratio decreases as the dropping position decreases. Although SPD's success lies in its ability to drop a packet and notify TCP early, dropping this packet too fast may give TCP a false alarm or cause TCP to slow down too fast. Thus, it is not recommended to set the SPD dropping threshold to be less than the 20-cell position. On the other hand, the simulation results show that if the dropping position is set to be greater than 20 cells, the goodput to input ratio decreases as the dropping position increases. This justifies that, at least for the present configuration, the SPD dropping position should be as early as possible, but not earlier than the 20-cell position.

5.3 Performance Comparison of EPDs and OSD for the Persistent Sources

After finding the optimal position for the single packet discard scheme, the performance by the optimal SPD[1] is compared to the optimal EPD[2] schemes of the same configuration. It is known that the performance of EPD is not directly proportion to the size of the EPD threshold. Thus, it is important to compare the performance of the optimal SPD scheme to that of the EPD schemes with a reasonable range of thresholds. The simulations are done for the persistent sources with 5, 15, 20, 25 VCs.

Two EPD mechanisms were compared to the SPD scheme. The first version of the EPD schemes is the aggregate EPD in which congestion status in different VCs do not contribute to the EPD packet-dropping decision. The second version of EPD scheme is the EPD per-VC. In this scheme, both the aggregate threshold and per-VC threshold are used. Hence, EPD per-VC is activated only when the switch buffer occupancy exceeds the aggregate threshold and the individual VC buffer occupancy exceeds the EPD Per-VC threshold.[3]

The results of this set of simulation are plotted in Figure 5.7 - 5.11 with exact values followed by each corresponding table.

1. Optimal SPD means that SPD scheme with the dropping threshold set to the optimal position, 20 cells.
2. Optimal EPD means that the best EPD goodput to input ratio among different EPD thresholds.
3. Per-VC threshold is defined as the aggregate threshold divided by the number of active VCs.

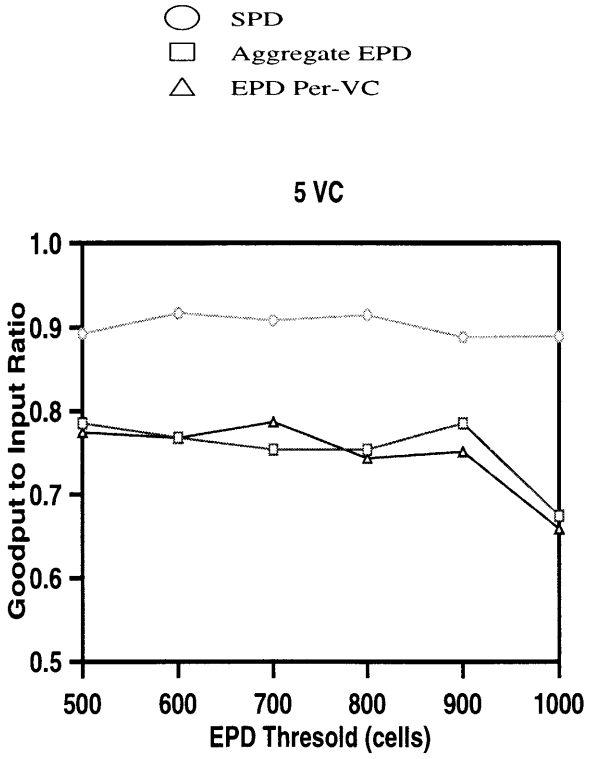

Figure 5.7: Goodput to Input Ratio versus EPD Threshold for 5 VCs

EPD Threshold (cells)	Goodput to Input Ratio for aggregate EPD	Goodput to Input Ratio for SPD	Goodput to Input Ratio for Aggregate and per-VC EPD
100	0.785	0.893	0.774
120	0.768	0.917	0.768
140	0.754	0.908	0.787
160	0.754	0.915	0.743
180	0.785	0.888	0.751
200	0.674	0.889	0.658

Table 5.7: Goodput to Input Ratio versus EPD Threshold for 5VCs

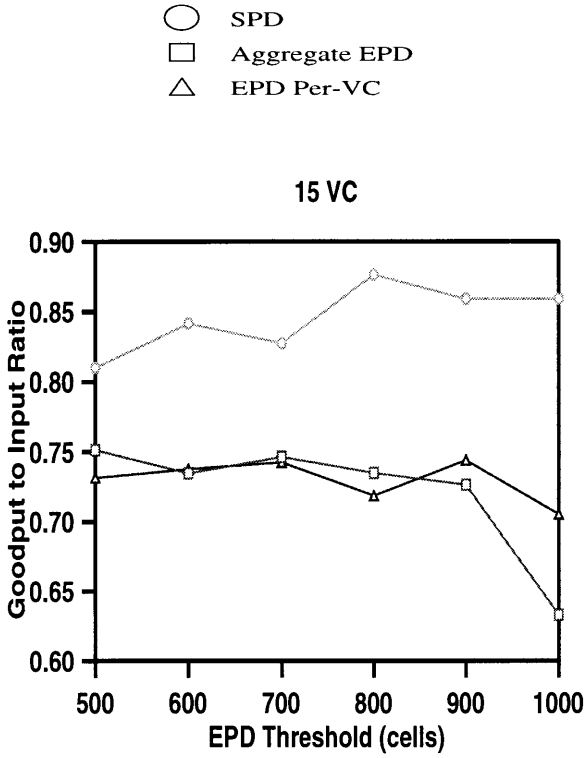

Figure 5.8: Goodput to Input Ratio versus EPD Threshold for 15 VCs

EPD Threshold (cells)	Goodput to Input Ratio for Aggregate EPD	Goodput to Input Ratio for SPD	Goodput to Input Ratio for Aggregate and per-VC EPD
500	0.751	0.810	0.731
600	0.735	0.841	0.738
700	0.746	0.827	0.742
800	0.735	0.876	0.719
900	0.726	0.859	0.744
1000	0.633	0.859	0.705

Table 5.8: Goodput to Input Ratio versus EPD Threshold for 15 VCs

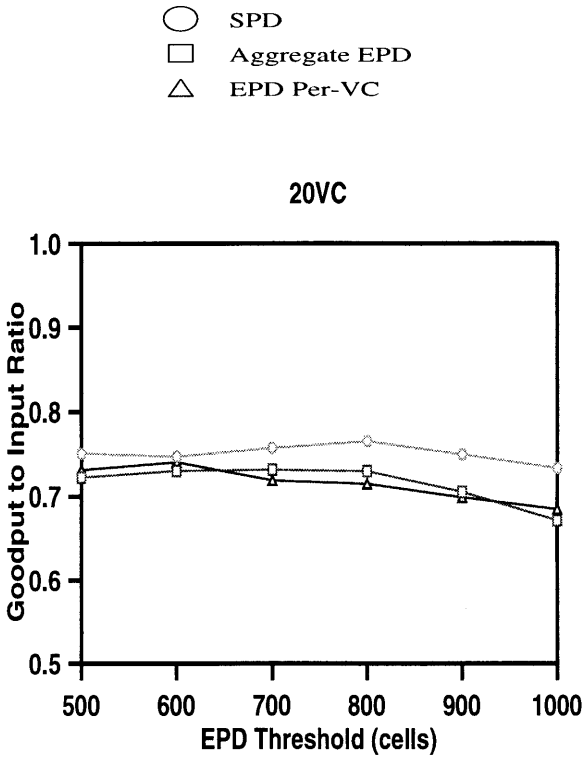

Figure 5.9: Goodput to Input Ratio versus EPD Threshold for 20 VCs

The EPD Threshold (cells)	Goodput to Input Ratio of Aggregate EPD	Goodput to Input Ratio for SPD	Goodput to Input Ratio for Aggregate and per-VC EPD
500	0.723	0.751	0.731
600	0.730	0.747	0.741
700	0.732	0.757	0.719
800	0.730	0.765	0.715
900	0.705	0.750	0.699
1000	0.671	0.733	0.685

Table 5.9: Goodput to Input Ratio versus EPD Threshold for 20 VCs

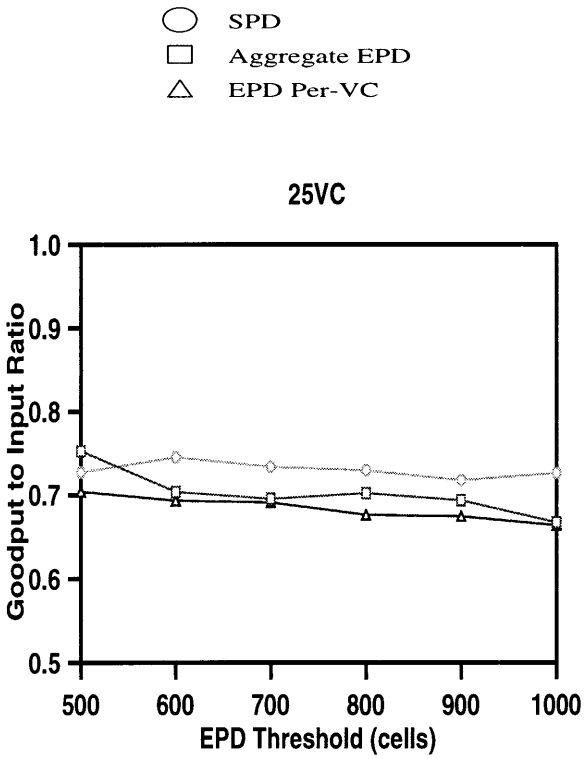

Figure 5.10: Goodput to Input Ratio versus EPD Threshold for 25 VCs

EPD Threshold (cells)	Goodput to Input Ratio for Aggregate EPD	Goodput to Input Ratio for SPD	Goodput to Input Ratio for Aggregate and per-VC EPD
500	0.752	0.727	0.704
600	0.703	0.745	0.693
700	0.695	0.733	0.691
800	0.702	0.729	0.676
900	0.693	0.717	0.674
1000	0.667	0.726	0.664

Table 5.10: Goodput to Input Ratio versus EPD Threshold for 25 VCs

As shown in Table 5.7 - 5.10, in most cases, the SPD Scheme gives the better performance for all different numbers of VC's. If the number of VCs is small, any SPD performance is higher than any other EPD performance regardless of the setting of the EPD threshold. For large number of VCs, the SPD performance is about same as the other EPD schemes.

This probably results from the fact that the SPD dropping threshold is close to the EPD thresholds when the number of VC is large. SPD performs better when there are at least 2 packets of buffer space between the SPD threshold and EPD threshold. This is shown in the results obtained from the simulations of 15 VCs. The large space can accept the incoming TCP packets before the TCP slows down the input rate. A few sets of simulations are run for SPD with larger buffer sizes and longer gaps between the SPD and EPD thresholds showing that the improvement is large.

Presently only the results from persistent sources with severe congestion are presented. However, it also important to find out how this scheme works with a more realistic set of data. The same set of simulations were then run with data of bursty nature and the results are discussed in the next section.

5.4 End-to-end Performance for Bursty Data Source

With bursty sources, each user sends out the packets according to the random Poisson arrival time described in Chapter 3. Thus, not all VCs are sending data all the time as opposed to persistent sources where users are all greedy. In this set of simulations, the result of 10 VCs is shown below.

Due to the bursty nature of this simulation, each value is averaged from five independent trials. For 10 VCs, μ is set to 0.00024. The average transmission rate for each VC is 0.24 Mbits/second.

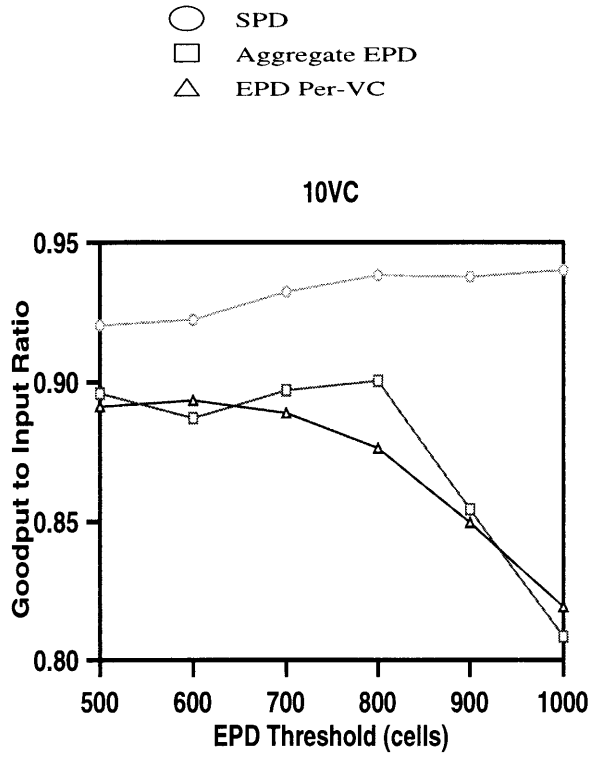

Figure 5.11: Goodput to Input Ratio v.s.EPD Threshold for 10 VCs Bursty Sources

EPD Threshold (cells)	Goodput to Input Ratio for Aggregate EPD	Goodput to Input Ratio for SPD	Goodput to Input Ratio for Aggregate and per-VC EPD
500	0.896	0.920	0.891
600	0.887	0.922	0.894
700	0.897	0.932	0.889
800	0.901	0.938	0.876
900	0.854	0.938	0.850
1000	0.808	0.940	0.819

Table 5.11: Goodput to Input Ratio v.s.EPD Threshold for 10 VCs Bursty Source

From the result in Figure 5.11 and Table 5.11, the goodput to input ratio drops as the EPD threshold goes from 800 to 1000. This is a consistent result as shown for persistent sources. The interesting result shows that as the EPD threshold increases, the general performance of the SPD does not decrease like that of EPD mechanisms. This might suggest that SPD scheme is a more stable scheme if the network is not as congested as in the case of persistent sources.

5.5 Comparing the Switch Queue for EPD and SPD schemes

Looking at the buffer queue for a congested switch helps with understanding how the mechanism works. As a means for discussion, Figure 5.12 shows the switch buffer queue versus time for a 15VC persistent source setting. The congested switch is switch 2 as indicated in Figure 4.1.

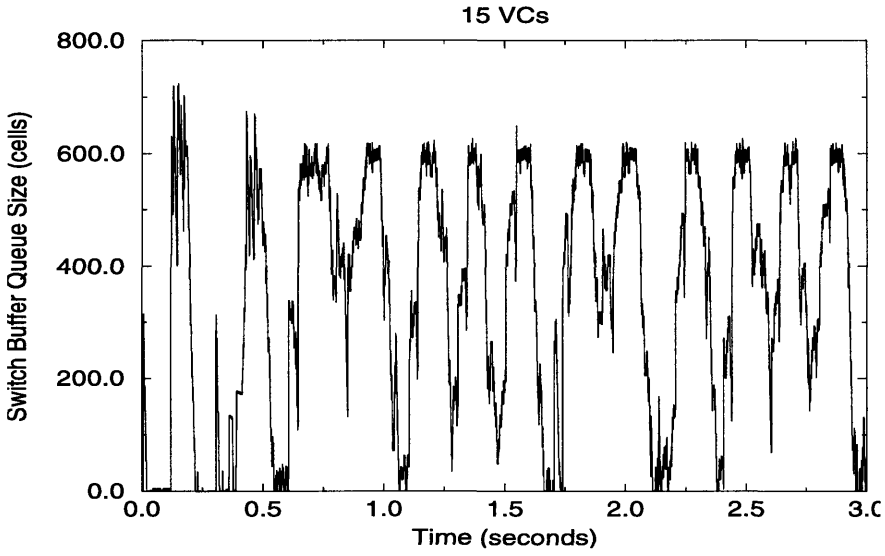

Figure 5.12: Switch Buffer Queue Size versus Time for Aggregate EPD Scheme

The simulation that generated Figure 5.12 - Figure 5.13 has an EPD factor of 0.6 and a maximum buffer size of 1000 cells. From Figure 5.12, it can be seen that each "peak" rep-

resents congestion. Until it reaches the EPD threshold, which is 600 cells, EPD starts dropping cells. The buffer stays at around 600 cells until TCP slows down its rate and congestion is alleviated. The buffer size decreases and reaches zero until it is congested again.

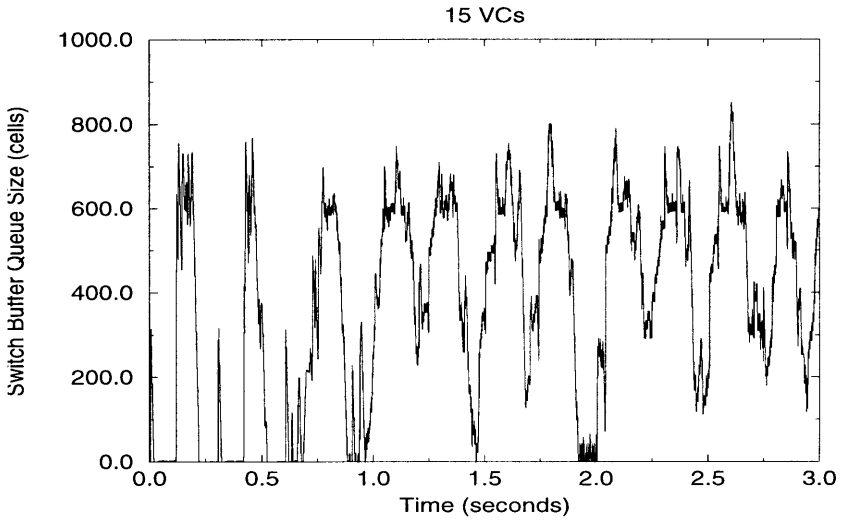

Figure 5.13: Switch Buffer Queue Size versus Time for EPD per-VC Scheme

The difference between aggregate EPD and EPD per-VC is that the peaks of the buffer size for the latter mechanism exceeds 600 cells. This happens when the buffer size reaches the 600-cell EPD threshold. Meanwhile, for the per-VC, the threshold is not yet reached, indicating that some incoming cells are still being received by the switch. The important thing to notice is the small horizontal black bands on the top of each "bump." These are good indications that many of the incoming cells were dropped during those times.

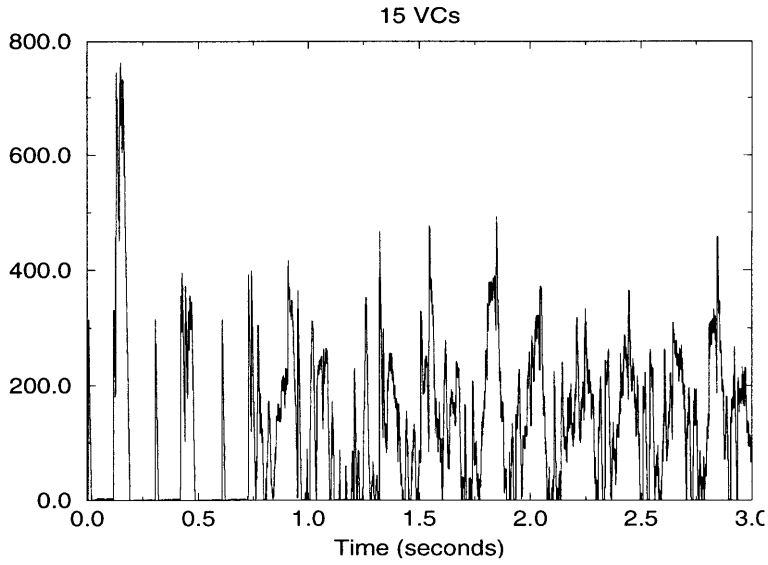

Figure 5.14: Switch Buffer Queue Size versus Time for EPD SPD Scheme

The SPD switch buffer queue plot has a different appearance than the EPD plots. In SPD, one cell is dropped early to notify TCP of the state of congestion every time the buffer started to build up from a size of zero cells. TCP is able to decelerate fast enough such that the build up of the buffer rarely reaches the EPD threshold. Thus, the dropping of the cells is not as severe as that of the EPDs mechanisms.

5.6 Overall Goodput Improvement

Previous sections in this chapter discussed and showed the improvement in the goodput to input ratio of the SPD scheme. The improvement in the goodput to input ratio suggests that bandwidth can be used more efficiently during congestion. If less bandwidth is wasted due to congestion, the preserved bandwidth can be allocated to other applications.

Another set of simulations are run to show that the overall goodput of SPD scheme improves from that of the EPD scheme with the same settings. The network topology is different than the peer-to-peer configuration described in Chapter 3. As shown from figure 5.15, there are two sets of peer-to-peer traffic. Originating from the persistent source users

at the top of the figure, 20 peer-to-peer connections pass through the Switch1 and the Switch2. 10 connections pass through Switch3 and 10 others pass through Switch4.

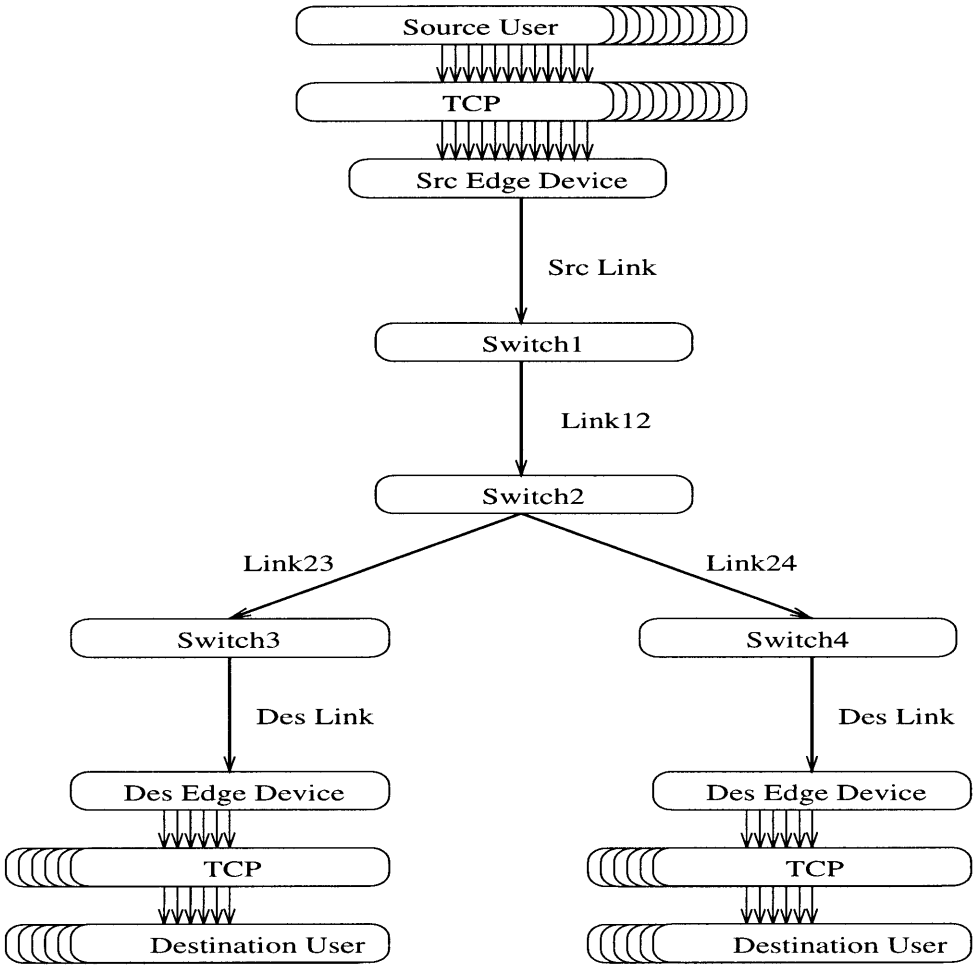

Figure 5.15: Network Topology for Goodput Improvement Simulations

The bandwidth is 80 Mbits/sec for Link12, 10 Mbits/sec for Link23, and 1 Gibbets/sec for Link24. Connections pass through Switch3 are congested in Switch2 because Link23 is the bottleneck for these connections. On the other hand, the connections that pass through Switch4 are congested at Switch1 because Link12 is the bottleneck. It is also found that if Link12 has much higher bandwidth than Link23, such as 150 cells, the bandwidth saved in Link23 is not significant enough for the other set of traffic.

The buffer size of each switch is set to 5000 cells while the SPD dropping threshold is set to 70 cells for Switch1 and Switch2. The SPD scheme is compared again to the aggregate EPD scheme. The EPD thresholds are: 0.5, 0.6, 0.7, 0.8, and 0.9. Figure 5.16 shows the sum goodput of both sets of connection using both SPD and EPD schemes.

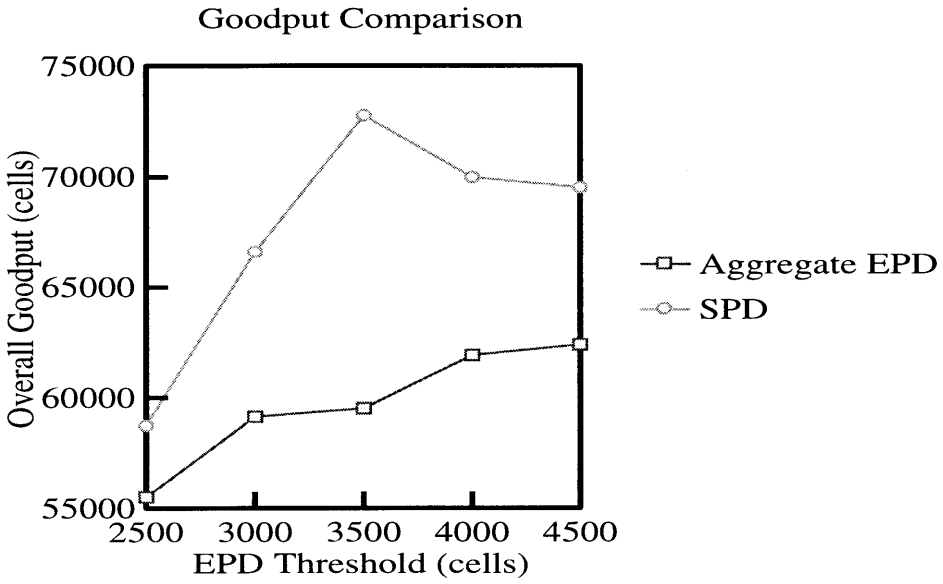

Figure 5.16: Goodput Comparison for Aggregate EPD and SPD schemes

EPD Threshold (cells)	Goodput for Aggregate EPD (cells)	Goodput for SPD (cells)	SPD Improve-ment percentage
2500	55514	58747	5.8%
3000	59144	66602	12.6%

Table 5.12: Goodput Comparison for EPD and SPD Schemes

EPD Threshold (cells)	Goodput for Aggregate EPD (cells)	Goodput for SPD (cells)	SPD Improve-ment percentage
3500	59519	72775	22.3%
4000	61918	69977	13.0%
4500	62396	69521	11.4%

Table 5.12: Goodput Comparison for EPD and SPD Schemes

The connections passing through Switch3 reach severe congestion in Switch2. When congestion first occurs, SPD quickly drops one packet at Switch2. The TCP of these connections reduce their rate once their timer expires. The input of these traffic is reduced and less bandwidth is used by these connections for Link12. The other set of connections are able to use this bandwidth which increases the overall sum goodput. From the values at Table 5.12, it can be shown that our the average goodput improvement is 13% for this configurations.

Chapter 6

Concluding Remarks

6.1 Summary

In this research, a packet-dropping technique-- the single packet discard-- is introduced as an improvement to the EPD schemes. SPD is studied and simulated in different network settings with both persistent and bursty sources. Unlike EPD schemes which could drop cells equivalent to multiple TCP packets, SPD only purposely drops cells equivalent to one packet size every time it detects a congestion.

SPD passes on the congestion status to TCP early enough such that the switch buffer never reaches the EPD threshold. This is accomplished because TCP window is reset to one packet when TCP detects any packet loss. When TCP reduces its window to a smaller size, it also decreases the input to the network. Improving the goodput to input ratio allows the network to use the available resources more efficiently. The saved resources can then be used by other end-to-end applications. This key SPD scheme can probably also be extended to other network protocols.

This thesis gives an overview of flow control schemes in TCP, ATM, and TCP over ATM interconnecting networks. The designed SPD scheme and the existing aggregate EPD and EPD per-VC schemes are simulated and the results are compared and discussed.

The simulation results can be summarized as followed:

(1) For switch buffer size of 1000 cells, the optimal dropping threshold is 20 cells for the range of active 15-40 VCs in the peer-to-peer network.

(2) The SPD generally gives a better goodput to input ratio for both persistent and bursty sources as long as it is given enough space (more than 2 packets in length) between

the dropping threshold and the per-VC threshold or the aggregate EPD threshold divided by the number of active VCs.

(3) SPD scheme can improve the overall goodput measurement. For the topology and parameters specified in section 5.6, an average of thirteen percent improvement is found.

6.2 Future Works

Although the goodput to input ratio for the SPD scheme is higher, the goodput might be compromised if the TCP window slows down too much or slows down too quickly. Future work will focus on expanding the SPD idea and applying it at the optimal time such that TCP is not slowed down too fast or too much. Perhaps, TCP Reno should also be considered [8].

The two parameters that are important for further exploration are:

1. The Switch Buffer Size. UBR generally works better with larger switch buffers. In Chapter 5, the results show that SPD works better with smaller number of VCs given the switch buffer size equals to 1000 cells. It is very likely that having a larger number of VCs can have a similar performance if the buffer size is proportionally larger.

2. The Roundtrip Delay of the network. SPD provides early TCP congestion notification. The early notification is more beneficial for end-to-end networks with long delays. Some might say that EPD works well with LAN while SPD works well with WAN.

References

[1] P. Newman, "Data over ATM: Standardization of Flow Control" SuperCon'96, Santa Clara, Jan 1996.

[2] N. Yin and S. Jagannath. End-to-End Traffic Management in IP/ATM Internetworks. ATM Forum Contribution 96-1406, October 1996.

[3] D. Comer and D. Stevens. Internetworking with TCP/IP. Prentice-Hall, Inc. Englewood Cliffs, New Jersey: 1991.

[4] The ATM Forum Technical Committee. Traffic Management Specification Version 4.0. ATM Forum aftm-0056.0000, April, 1996.

[5] W. Ren, K. Siu, H. Suzuki and G. Ramamurthy. Performance of TCP over ATM with Legacy LAN. ATM Forum, December 1996.

[6] P. Narvaez and K. Siu. An Acknowledgment Bucket Scheme for Regulating TCP Flow over ATM. To appear in IEEE Globecom '97.

[7] A. Romanow and S. Floyd "Dynamics of TCP Traffic over ATM Networks," *Proceedings of SIGCOMM 1994*, August 1994, pp 79-88.

[8] R. Stevens. TCP/IP Illustrated, Volume. 1, Addison-Wesley Publishing Company, Reading, MA: 1994.