THE UNIVERSITY OF CALGARY

The Performance of TCP over ATM on Lossy Asymmetric Networks

by

Guang Lu

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

April, 2000

0-612-55226-8

Abstract

This thesis studies the end-to-end performance of the Transmission Control Protocol (TCP) over Asynchronous Transfer Mode (ATM) when the access network is noisy and asymmetric. In particular, the Asymmetric Digital Subscriber Line (ADSL) access network is examined. TCP experiences performance degradation in this network architecture because of the overhead of protocol conversion and the data loss due to transmission errors.

A simulation model is developed to simulate ADSL network components and noisy local loops. Several groups of experiments are conducted in order to explore TCP performance under various error ratios in different network scenarios. Experimental results illustrate how TCP is affected by errors along with other factors, such as TCP Maximum Segment Size (MSS), bandwidth asymmetry, the percentage of noisy lines, and buffer overflow by congestion. Observations are made that suggest possible methods to improve TCP performance. For example, when cell error rates are high, TCP achieves higher throughput in the presence of burst errors than in the presence of randomly scattered errors, and a smaller MSS makes TCP more resilient to noise.

Acknowledgments

I would like to give my sincere thanks to Dr. Brian Unger, my supervisor, for his guidance and support throughout my study at the University of Calgary. But for his bringing me into this area, I would not be able to work on what I am interested in. His comments and suggestions have greatly helped me to improve my thesis quality. He has provided significant support in bringing this thesis in completion.

I am so grateful to Dr. Carey Williamson. He has given me tremendous help in my thesis research, especially in the experimental design and results analysis. I would not have been able to finish the thesis without his help and his expertise.

I want to express my deep gratitude to Zhonge Xiao for his help in every stage of this project, from the problem definition to thesis reviewing. It is hard to imagine doing this work without his help.

I owe great thanks to Rob Simmonds for his continuous help in every aspect of my study. He makes the work so enjoyable with his intelligence and great sense of humor.

Thank Dr. Saul Greenberg for being on my committee.

Thanks also go to the past and present members in the Telesim project team for providing a friendly and stimulating work environment. It is really a pleasure to work with such a group of intelligent and dedicated people. Specially, I am thankful to the peer student and my friend Jiayun Zhu for all the help and being a companion during my stay in Calgary. Thank Mike Bonham for proofreading my thesis and helping me bind the thesis.

I appreciate the help from the staff and students in the Computer Science Department at the University of Calgary. I also want to thank for the assistance and facility I got from the MACI project.

I am thankful to all my friends who have been so supportive.

I am indebted to a lot of people who have directly or indirectly helped me, although I cannot list all their names here.

Finally, I want to thank my parents for their love and encouragement in all my endeavors. I also thank my sister and all the relatives for their support. I dedicate this thesis to my family across the miles.

Contents

List of Tables

List of Figures

List of Abbreviations

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACK	Acknowledgment
ADSL	Asymmetric Digital Subscriber Line
AN	Access Node
ATM	Asynchronous Transfer Mode
ATU-C	ADSL Transmission Unit-Central Office
ATU-R	ADSL Transmission Unit-Remote Terminal
B-ISDN	Broadband Integrated Services Digital Network
CBR	Constant Bit Rate
CER	Cell Error Ratio
CLR	Cell Loss Ratio
CO	Central Office
DES	Discrete-Event Simulation
DSLAM	Digital Subscriber Line Access Multiplexer
ECN	Explicit Congestion Notification
EPD	Early Packet Discard
FEXT	Far-End Crosstalk

FEC	Forward Error Correction
FTP	File Transfer Protocol
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU-T	International Telecommunications Union Telecommunication Standardization Sector
LAN	Local Area Network
LP	Logical Process
MPEG	Motion Pictures Expert Group
MSS	Maximum Segment Size
MTU	Maximum Transfer Unit
NEXT	Near-End Crosstalk
nrt-VBR	Non-Real-Time Variable Bit Rate
PDN	Premises Distribution Network
POTS	Plain Old Telephone Service
PPD	Partial Packet Discard
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
PVC	Permanent Virtual Connection
QoS	Quality of Service
RED	Random Early Detection
rt-VBR	Real-Time Variable Bit Rate
RTT	Round-Trip Time

SMTP	Simple Mail Transfer Protocol
STM	Synchronous Transfer Mode
SVC	Switched Virtual Connection
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Terminal Equipment
UBR	Unspecified Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier
WAN	Wide Area Network
WWW	World Wide Web

Chapter 1

Introduction

The Internet has been growing explosively during the past decade. The World Wide Web (also called the Web or WWW) is the fastest growing application on the Internet. Internet ubiquity and Web explosion have led to dissatisfaction with the speed and efficiency of the Internet. Until recently, Internet access has been by low-speed dial-up telephone lines. This speed cannot meet the increasing requirements for fast Internet access, especially with the spread of multimedia applications. There has been a continual and growing demand for broadband services in local loops.

Asymmetric Digital Subscriber Line (ADSL) is a copper loop transmission technology that was designed to provide the broadband solution for residential and small business users. It is built upon the existing architecture of Public Switched Telephone Network (PSTN). ADSL is currently being deployed as one of the broadband local loop approaches. Other approaches include cable modems and satellite systems.

The Internet architecture is a layered design. The highest layer is the application layer including protocols such as the File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP). The next layer is called the transport layer. Transmission Control Protocol (TCP) is the protocol at this layer. The layer below is the network layer. Internet Protocol (IP) resides in this layer. TCP/IP is the dominant protocol suite for the Internet. The lowest layer is the physical/link layer. Asynchronous Transfer Mode (ATM) has been used at the physical/link layer. ATM is a multiplexing and switching technology that enables the integration and fast-speed transport of data, voice, and video. ATM is often adopted by service providers in their backbones for high-speed Internet services. ADSL is the physical layer technology on access networks. This layered network architecture is called TCP over ATM over ADSL and denoted as TCP/ATM/ADSL in the thesis.

The thesis addresses the performance issues of TCP over ATM over ADSL. In the following sections of this chapter, the relevant background of telecommunications is introduced, the goals of the research are listed, and the outline of the thesis is drawn.

1.1 TCP/ATM/ADSL

In this research, a study of the end-to-end performance of TCP over ATM with ADSL technology implemented on the access network is conducted. The main features of TCP, ATM, ADSL, and the potential problems in the architecture of TCP/ATM/ADSL are described in the following subsections.

1.1.1 Transmission Control Protocol (TCP)

TCP is the main protocol that is used as the transport layer protocol on the Internet. It was designed to provide reliable end-to-end data transportation over unreliable networks [18]. Standard applications based on TCP include the File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Telnet terminal services.

TCP is a connection-oriented protocol. Both hosts in one TCP conversation make use of an interface called a *socket*. A logical connection must be explicitly established between the socket of the sender and the socket of the receiver [32]. TCP transmits data in the form of segments (sometimes called packets). A *segment* consists of a 20-byte header followed by zero or more data bytes. The segment header contains important information for data delivery, such as source and destination ports, sequence number, acknowledgment number, window size, and checksum. Each segment is acknowledged and the missing segments are retransmitted so that TCP ensures reliable data delivery.

TCP uses a sliding window mechanism for flow control. There are four intertwined TCP congestion control algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. The slow start and congestion avoidance algorithms are used to control the amount of data being injected into the network. The fast retransmit and fast recovery algorithms are used to trigger retransmissions sooner than the regular timeout mechanism when congestion occurs. The details of these algorithms will be introduced in Subsection 5.1.1.

There are different versions of TCP implementations. The most well-known ones are TCP-Tahoe, TCP-Reno, TCP-NewReno, and TCP-Vegas. TCP-Tahoe implements the slow start and congestion avoidance algorithms. TCP-Reno includes the fast retransmit and fast recovery algorithms in addition. TCP-NewReno modifies the fast retransmit mechanism. TCP-Vegas is a new implementation of TCP. It achieves better throughput than the old implementations by modifying the slow start, congestion avoidance, and retransmission mechanisms.

TCP is able to provide robust end-to-end performance in a wide variety of network environments from Local Area Networks (LANs) to Wide Area Networks (WANs). However. emerging networking technologies pose new challenges to TCP in terms of performance. The research work that has been done on TCP performance will be reviewed later in Chapter 2.

1.1.2 Asynchronous Transfer Mode (ATM)

ATM is the transport mode of choice for Broadband Integrated Services Digital Network (B-ISDN). B-ISDN is the broadband extension to the fixed bandwidth digital integration services called Integrated Services Digital Network (ISDN). The idea of ISDN is to provide a technique to transmit integrated digitized data, voice, and other applications using the existing copper subscriber loops. B-ISDN is targeted to provide high bandwidth, bandwidth on demand, various Quality of Services (QoS), connection-oriented or connectionless, constant and variable bit rate services on the same network [26].

ATM is the transfer mode choice for B-ISDN by the International Telecommunications Union Telecommunication Standardization Sector (ITU-T). ATM is connection-oriented so that end-to-end connections must be established before data is transmitted. ATM connections are either preestablished using management functions or dynamically set up on demand using signaling [26]. Preestablished connections are referred to as Permanent Virtual Connections (PVC), whereas dynamically set up connections are referred to as Switched Virtual Connections (SVC). One or more virtual connections can run over one physical link.

ATM uses fixed-length packets that are referred to as ATM *cells*. A cell is 53 bytes long with 48 bytes of payload and 5 bytes of header. The size of 53 bytes is a compromise that allows the integration of different services with different characteristics and requirements. ATM is characterized as *asynchronous* because cells can be transmitted from a source to a destination at irregular time slots. ATM uses a packet-switching technology to transport cells. A cell header contains a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI). Cells are switched according to the VPI/VCI table in ATM switches.

The ATM architecture consists of the physical layer, the ATM layer, and the ATM adaptation layer (AAL) [16]. The physical layer is responsible for the transmission of cells between two ATM hosts over the physical medium. The ATM layer is responsible for flow control, cell header generation, multiplexing and demultiplexing, and VPI/VCI translation. The AAL maps higher level services into the ATM layer. Different AAL service classifications are defined to match different service classes [17].

ATM supports different Quality of Services (QoS), which means that ATM will provision resources to guarantee a specified minimum throughput, maximum delay, and maximum data loss for the duration of a particular connection [29]. This feature enables ATM to support different kinds of traffic, such as data, voice, and video by providing different QoS parameters for them. The ATM Forum has defined five service categories. They are Constant Bit Rate (CBR). Real-Time Variable Bit Rate (rt-VBR), Non-Real-Time Variable Bit Rate (nrt-VBR). Unspecified Bit Rate (UBR). and Available Bit Rate (ABR) [3].

1.1.3 Asymmetric Digital Subscriber Line (ADSL)

ADSL is a modem technology for delivering broadband services to homes and small businesses. It is built upon the existing twisted copper pair telephone lines. An ADSL modem works by transmitting Internet data in a frequency range that is separate from the 4 KHz range used for voice transmission. thus supporting simultaneous voice and data communications over one copper line. ADSL can support a wide variety of high bandwidth applications. such as high speed Internet access. telecommuting. and video-on-demand. ADSL is *asymmetric* in that it provides a higher bandwidth for downstream traffic than for upstream traffic. This can be sufficient and efficient because many Internet-based user activities are inherently asymmetric (e.g.. Web document downloads).

The major ADSL components include ADSL Transmission Unit-Remote Terminal (ATU-R). ADSL Transmission Unit-Central Office (ATU-C). and Digital Subscriber Line Access Multiplexer (DSLAM). Each traffic source is connected to an ATU-R on the remote side and an ATU-C on the central office side. Traffic streams are multiplexed by the DSLAM and routed to different Internet Service Provider (ISP) networks. ATU-Cs and the DSLAM form the Access Node (AN) to the core networks. The details of ADSL technology will be introduced later in Chapter 2.

Since the transmission media for ADSL are twisted copper pairs. transmission errors are inevitable. The factors that cause errors include crosstalk, impulse noise, and attenuation. The line's physical attribute (attenuation) along with impairment (crosstalk) is critical to a digital modem's overall performance, thus affecting the end-to-end performance of the network. The impact of errors on the end-to-end network performance has been studied intensively in this work and will be described in the following chapters of the thesis.

ADSL is a technology that has attracted much attention and interest. The main advantages of ADSL are that it:

- provides broadband services to home users and small businesses;
- builds on existing telephone lines;
- takes full advantage of existing network protocols;
- supports simultaneous transmission of voice and data over one single telephone line.

1.2 Performance Issues for TCP/ATM/ADSL

ADSL systems need to support multiple services, such as frame relay, ATM, and Local Area Network (LAN), for the purpose of interoperability. The ADSL Forum has recommended ATM over ADSL as a standard model. This recommended model was studied using TCP as the transport layer protocol, which is denoted as TCP/ ATM/ADSL in the thesis. The ATM over ADSL architecture preserves the high-speed characteristics of ATM and ADSL, and guarantees QoS support. However, there are some performance problems that need to be addressed in the TCP/ATM/ADSL architecture.

First, TCP experiences performance degradation on ATM networks. The performance problems are primarily due to the segmentation and reassembly process required for TCP/IP packets on ATM networks [15]. That is, the ATM AAL layer breaks each TCP segment into ATM cells at the source for transmission over the ATM network, and reassembles these cells into a TCP segment at the destination. The loss of one cell will cause the loss of the whole TCP segment and result in retransmission. The performance of TCP over ATM and methods to optimize the performance have been, and remain an interesting phenomenon to study.

Second, the network asymmetry affects TCP performance because TCP relies on feedback in the form of cumulative acknowledgments from the receiver to ensure reliability [4]. TCP relies on the timely arrival of acknowledgments to progress and fully utilize the available bandwidth. Therefore, any disruption in the feedback will impair the performance.

Third, errors affect the performance of TCP/ATM/ADSL. ADSL was invented to utilize the existing telephone lines for high speed digital data transmission. However, the traditional twisted copper pairs are error-prone. Both internal and external factors can cause transmission errors on the copper pairs. Errors on the lines will cause the loss of TCP segments and force TCP retransmissions, thus reducing the overall TCP performance.

More discussion on the TCP performance and unresolved problems associated with ADSL will be given in Chapter 2.

1.3 Thesis Goals

The objective of this research is to study the performance of TCP over ATM on a network where ADSL technology is applied in the local loop. The approach of the study is to build a simulation model that represents the major characteristics of ADSL. This ADSL simulation model is embedded in an existing ATM simulator called the ATM-TN [33], which stands for Asynchronous Transfer Mode Traffic and Network. The emphasis is given to the errors on ADSL lines and their impact on the end-to-end TCP performance.

As a synopsis, the goals of this research are to:

1. describe the network architecture of ATM over ADSL;
2. design and implement a simulation model of ADSL in the ATM-TN;
3. design and conduct experiments to evaluate the performance of TCP over ATM over ADSL under various network conditions by setting up different parameters, including the cell error rate, TCP segment size, ADSL line bandwidth, percentage of lines that are noisy, and ATM switch buffer size;
4. analyze the experimental results and explore the characteristics of TCP under the effects of transmission errors, network asymmetry, and congestion.

1.4 Thesis Structure

Chapter 2 reviews the research work that has been done on relevant subjects, such as TCP performance under various network configurations. It also provides more information on the ADSL technology and discusses some unresolved issues of ADSL. Chapter 3 presents the design issues, the architecture of the ADSL model, and the development of the simulation program. Chapter 4 introduces the experimental design, the traffic model, and the performance metrics. Chapter 5 describes the experiments that have been done, analyzes the results, and evaluates the performance. Finally, Chapter 6 concludes this thesis with a summary of the work that has been done, the contributions, and suggestions for future research.

Chapter 2

Literature Review

The study of TCP over ATM over ADSL (TCP/ATM/ADSL) is complex for two primary reasons. One is the dynamics of TCP on ATM networks. The other is the unknown characteristics and performance of ATM over ADSL and TCP over ADSL. This chapter addresses the above problems with a review of the related research that has been done on TCP performance. The features of ADSL are also studied.

Section 2.1 focuses on TCP performance over ATM networks, under random packet loss, and on asymmetric networks. Section 2.2 introduces ADSL access network elements, characteristics, and architecture. Finally, Section 2.3 discusses some open questions of ADSL and reiterates the purpose of this study.

2.1 TCP Performance

Although TCP has robust performance over a broad range of network architectures, it experiences performance degradation under certain circumstances. This section surveys the performance of TCP in several network environments. Subsection 2.1.1 examines TCP performance over high speed ATM networks. Subsection 2.1.2 describes TCP behavior over unreliable transmission media. Subsection 2.1.3 studies the effects of asymmetry on TCP performance. Please refer to Subsection 1.1.1 for a brief introduction to TCP and Subsection 5.1.1 for the detailed description of TCP congestion control algorithms.

2.1.1 Over ATM Networks

ATM provides the possibility for high speed TCP networking with very low transmission error ratios. However, experiments with TCP over ATM revealed poor performance under certain circumstances.

The performance degradation of TCP over ATM is primarily due to the segmentation and reassembly process required for ATM networks [15] [22]. Since TCP uses various size packets and ATM uses fixed size cells, the ATM Adaptation Layer (AAL) protocol is required to break up each TCP packet into ATM cells at the source for transmission on the ATM network, and to reassemble these cells into a TCP packet at the destination. This process cannot succeed unless all the cells that belong to the same TCP packet have been successfully received. The loss of a single cell means that the entire TCP packet is lost and must be retransmitted by the sender. Simulation results in [28] show that the effective throughput of TCP over non-ATM networks is higher than the effective throughput of TCP over plain ATM with the same network scenario parameters when cell loss occurs.

Cell loss on ATM networks is caused mainly by buffer overflows when the network is congested. The transmission media for ATM are typically very reliable optical fibers so that errors are negligible. The primary reason for low performance is that when cells are dropped at the switch, the congested link still transmits other cells from "corrupted" packets through the network. A "corrupted" packet is a packet that has at least one cell lost. Sending useless cells worsens the congestion and reduces the transmission of good data.

The mismatch of the size of TCP packets and ATM cells implies that larger packet sizes result in poorer TCP performance. Some research revealed that the overall effective throughput decreased when the TCP packet size increased [15] [28]. It is intuitive that the network overhead for retransmitting a smaller packet is lower [15].

Although the poor performance of TCP over ATM networks is mainly caused by the inevitable process of fragmentation and reassembly, the situation can be ameliorated. One way to improve TCP performance over ATM is to apply Partial Packet Discard (PPD) and Early Packet Discard (EPD) congestion control mechanisms. PPD drops all subsequent cells from a packet as soon as one cell has been dropped. EPD drops packets whenever the switch buffer exceeds a fixed threshold, thus preventing congestion and transmission of corrupted packets. Simulation studies show that PPD offers limited effective throughput improvement because cells are dropped when the buffer overflows, so that the congested link still transmits a significant fraction of cells belonging to corrupted packets. EPD achieves higher effective throughput by dropping complete packets when congestion is imminent [28].

Another way to improve TCP performance is to increase buffer sizes. Simulation experiments were performed to study the effect of buffer size on the aggregate effective throughput [15] [28]. As expected, effective throughput improves when buffer size increases. It is straightforward that a large buffer can accommodate more traffic bursts. Although the performance increase is gained at the expense of longer delays in the buffer, i.e., larger latency, the delay is negligible compared with the delay caused by retransmissions.

2.1.2 Over Lossy Networks

Noise is an important factor that causes packet loss in data communications. Types of noise include crosstalk, attenuation, impulse noise, and radio-frequency impairment. When operating on a network that is prone to transmission errors, TCP reacts in the same way as it does during congestion by slowing down transmissions. This results in a sharp decrease of TCP performance.

Simulation studies show that TCP is more sensitive to the loss of data than to the loss of acknowledgments (ACKs) [7]. Under the same loss ratio, data loss causes more severe throughput decrease than ACK loss does. Each data packet loss results in one or more data packet retransmissions. TCP shrinks its current transmission window under the occurrence of data loss. The performance degrades because TCP operates with a smaller window size than the permissible size and does not utilize the bandwidth efficiently.

Experiments show that TCP can hardly recover from packet loss when multiple packets are lost in the same transmission window [7]. Usually two consecutive packet losses can affect TCP performance seriously. Although the transmitter can recover from the first packet loss using fast retransmit, TCP's transmission window is usually not large enough for the duplicate ACKs to trigger another fast retransmit to recover from a second packet loss.

The performance of different versions of TCP is affected to different extents under conditions of random packet loss [24] [25]. TCP-Tahoe performs the worst in the presence of random loss. The fast retransmit feature in TCP-Reno and TCP-NewReno improves performance significantly. TCP-Tahoe is more robust than TCP-Reno in dealing with phase effects and multiple packet loss.

There are ways to tune TCP for higher performance. One way to increase the robustness of TCP on lossy networks is to use a smaller Maximum Segment Size (MSS). TCP can detect packet loss faster with a smaller MSS because more ACKs can be held in the TCP window. Using a finer granularity retransmission timer than the 500 ms timer used in most current implementations will enable TCP to react more quickly to packet loss. A "super fast retransmit" mechanism is proposed in [7]. Instead of having three duplicate ACKs to indicate a packet loss, receiving one duplicate ACK will trigger a fast retransmit. This mechanism works well on networks where TCP packets are guaranteed to be received in order because the receipt of the first duplicate ACK is a clear signal that a data packet has been lost.

2.1.3 Over Asymmetric Networks

Asymmetry is an inherent phenomenon on ADSL access networks. A network is said to exhibit asymmetry if the throughput achieved is not solely a function of the link and traffic characteristics in one direction, but depends significantly on those of the reverse direction as well [4]. The types of asymmetry include bandwidth asymmetry, latency asymmetry, and error ratio asymmetry. Network asymmetry affects the performance of TCP because TCP relies on timely feedback in the form of cumulative acknowledgments from the receiver to ensure data reliability. Disruption in the feedback can impair the performance of data transfer.

Bandwidth asymmetry impairs TCP performance in that it slows down the growth of the TCP window size. Consider two data packets transmitted by the sender in quick succession on a network with a higher bandwidth for data transfer and a lower bandwidth for acknowledgments. These packets are spaced apart according to the bottleneck link bandwidth in the data transfer path. The principle of ACK clocking is that ACKs preserve this spacing all the way back to the sender, enabling it to clock out new data packets [4]. However, ACKs may get queued because of the low bandwidth. Therefore, the spacing between the packets is dilated with respect to their original spacing when ACKs are queued. The consequence of this is that the sender's window growth is slowed down.

Latency asymmetry impairs TCP performance by varying the round trip time. Ideally, the round trip time should be relatively constant. Unfortunately, the round trip time can experience a high variation on a network with latency asymmetry. This can result in long idle time and slow reaction to packet loss, causing TCP performance degradation.

Two solutions are proposed in [4] to alleviate the effect of bandwidth asymmetry on TCP performance. One solution is called ACK Congestion Control (ACC). The approach to ACC is to use the Random Early Detection (RED) algorithm. It detects congestion by tracking the average queue size in the recent past. If the average queue size exceeds a threshold, a packet is selected randomly and marked with Explicit Congestion Notification (ECN). Upon receiving a packet marked with ECN, the sender reduces its sending rate. Another solution is called ACK Filtering

(AF). When an ACK from the receiver is about to be enqueued, the router checks its queue for any previous ACKs belonging to the same connection and removes them. The policy to filter the ACKs can be deterministic or random. In this way more space is made available for ACKs.

Three schemes for improving TCP performance over asymmetric networks are proposed and studied in [21]. One scheme is to provide higher priority to ACKs waiting for transmission on the slow link. The drawback of this scheme is that the throughput improvement is achieved at the expense of the slow connection. The second scheme is to limit the number of data packets in the outbound queue for the slow link. Although it improves the TCP throughput, this scheme results in an undesirable sensitivity of the two connections to each other's parameters, such as packet and ACK sizes. The third scheme is to use a connection-level bandwidth allocation mechanism. In a simple counter-based implementation, it counts the total number of bytes transmitted in a sequence of ACKs, and forces the transmission of a data packet if that number exceeds a preset threshold representing the desired ratio of data to ACK bandwidth. The advantage of this scheme is that it improves the fast connection's efficiency by properly controlling the bandwidth allocation between data packets and acknowledgments over the slow link.

2.2 An Overview of ADSL

This section presents an overview of the ADSL technology. Subsection 2.2.1 introduces ADSL components and architecture. Subsection 2.2.2 describes how the higher level protocol data packets are transformed and carried from end to end by

ADSL. Since ADSL is just a technology at the physical layer, it is necessary to have a whole network protocol stack built upon it. ATM and TCP are two of these protocols that are most likely to be in the architecture. Subsection 2.2.3 shows how to implement TCP over ATM over ADSL.

2.2.1 ADSL Network Components

Figure 2.1 illustrates the essential elements of ADSL in the network architecture. ADSL Transmission Unit-Remote Terminal (ATU-R) is the remote modem that connects customer site equipment to ADSL loops. ADSL Transmission Unit-Central Office (ATU-C) is the counterpart of ATU-R in the Central Office (CO), which is usually integrated as a single unit with the Digital Subscriber Line Access Multiplexer (DSLAM). DSLAM is the cornerstone of the ADSL solution. DSLAM is basically a multiplexer that concentrates the data traffic from multiple ADSL loops into the backbone network for connection to the service networks. DSLAM is the host of ATU-Cs and is usually built with an ATM access switch or IP router to provide multiservices. Key features of DSLAM include multiservices support, ADSL line aggregation, ADSL line code support, network management, etc. ATU-Cs, DSLAM, and the access equipment (ATM access switch, IP router, etc.) form the Access Node (AN) between the core network and the access network [8] [14].

Figure 2.2 provides a closer view of the ADSL access network and demonstrates how it handles different traffic streams. The wiring to existing analog telephones should not need to be altered for broadband services because a special splitter can separate these analog signals at the customer site. In the central office, the analog voice service is passed to the PSTN voice switch by another splitter. The ADSL

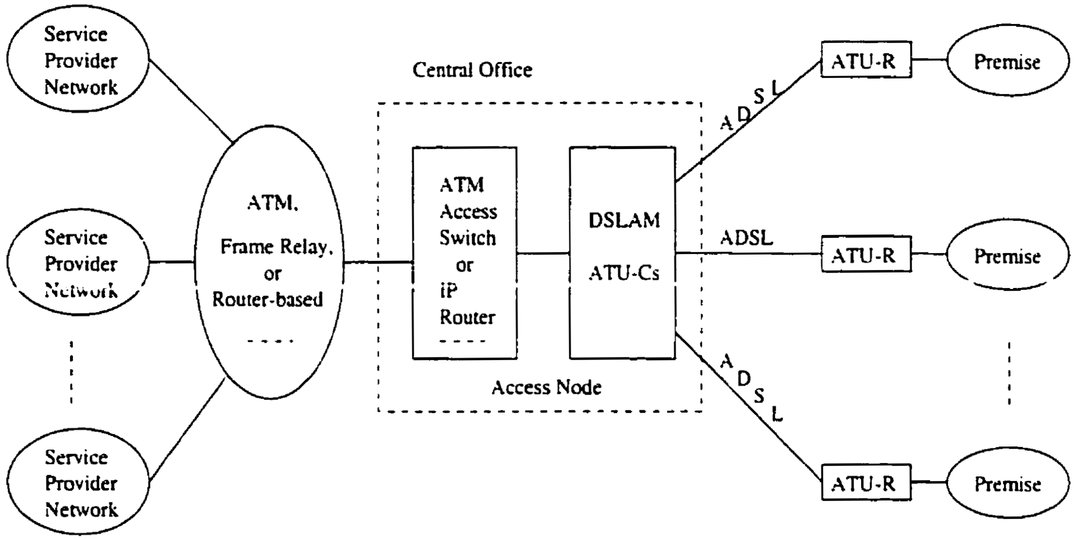


Figure 2.1: ADSL-Based Broadband Network Architecture

local loops are terminated and multiplexed in the DSLAM before leading to the CO switch.

2.2.2 ADSL Transport Modes

The ADSL Forum has defined five transport modes for ADSL as illustrated in Figure 2.3 [9][14]. The transport mode determines the format of ADSL frames when they are sent. The different distribution modes decide the different network architecture based on them. Figure 2.3 presents a simplified picture of a full ADSL network and the five transport modes.

The first mode uses Synchronous Transfer Mode (STM) from end to end. The ADSL network is a passive bit pump and provides only Time Division Multiplexing (TDM) and a constant bit rate on the established ADSL channels.

The second mode has packets flowing through the network. It is likely that the packets are IP packets, or they can represent other protocols so long as both

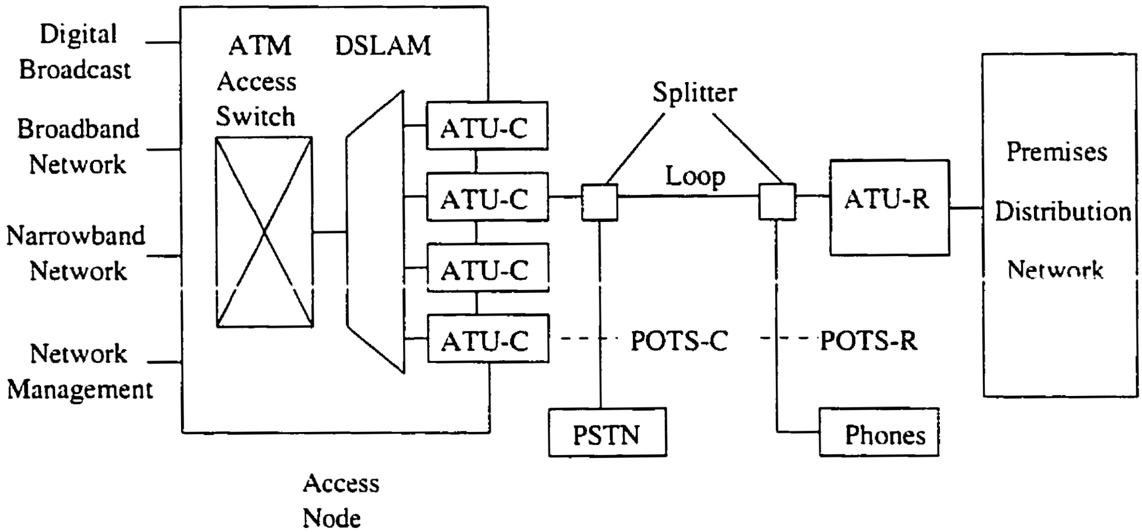


Figure 2.2: ADSL System Reference Model [9]

ends understand the format. The network becomes a more active partner in this mode because it can combine flows of packets and switch them to and from various end points. This transport mode enables the network architecture of end-to-end TCP/IP over ADSL.

The third mode is a combination of bit sync and ATM cells. The ADSL Access Node still handles bit streams on the ADSL access network, but converts bits to ATM cells over the service networks. This approach requires ATM capability in the ADSL Access Node. The advantage of this mode is that service providers can utilize the ATM backbones that they have deployed while not requiring users to adopt ATM.

The fourth mode uses packets on the ADSL links and ATM cells on service networks. This mode takes advantage of ATM in the backbone, and at the same time, avoids the passive bit pumps on access networks compared with the third mode.

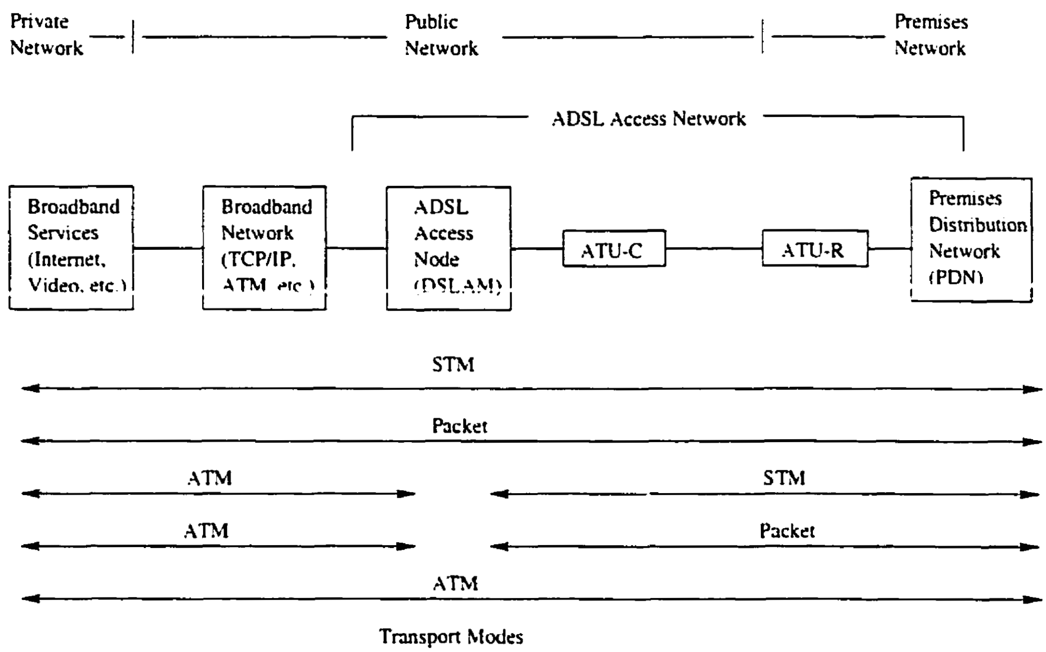


Figure 2.3: ADSL Transport Modes [9] [14]

Finally, the fifth mode employs ATM from end to end. There is a flow of cells transmitted between end points on the networks. In this mode, ATU-R and ATU-C must have the ATM adaptation function. Cells are multiplexed by the DSLAM and switched to different service networks. The contents of these ATM cells may still be IP packets. This ATM mode has evoked the greatest amount of interest among ADSL vendors and service providers. A substantial percentage of ADSL networks used this mode, especially for corporate customers or those who already have ATM equipment on the premises. Recently the ADSL Forum has decided to adopt Point-to-Point Protocol (PPP) over ATM over ADSL as a standard. The ATM end to end transport mode is modeled and studied in this thesis.

2.2.3 TCP/ATM/ADSL

Since ADSL was invented mainly for fast Internet access from customer premises, TCP/IP is the major protocol suite running over ADSL networks. The debate is about what could be the underlying protocols to support TCP/IP. The ADSL Forum has recommended Point-to-Point Protocol (PPP) over ATM as a standard for ADSL.

Figure 2.4 illustrates the network architecture of TCP over ATM over ADSL from customer premise networks to service networks. The ATU-R is used as the ATM access device. It has the ATM adaptation functions and routing functions. A PPP session runs on top of ATM Adaptation Layer 5 (AAL5) and is terminated at the backbone edge device using a PPP server with an ATM interface. ATM traffic is integrated into the DSLAM at the central office. The core network is the ATM switching network that switches traffic to different service networks.

The advantages of this network architecture are as follows. First, it utilizes the large deployment of ATM in network backbones. Second, ATM enables full service ADSL by integrating voice, video, and data into the same physical network. Third, for those who need compliance with B-ISDN, ATM is a good choice because it supports B-ISDN standards. Fourth, by using ADSL at customer premises, it avoids running fiber optics to every home and office.

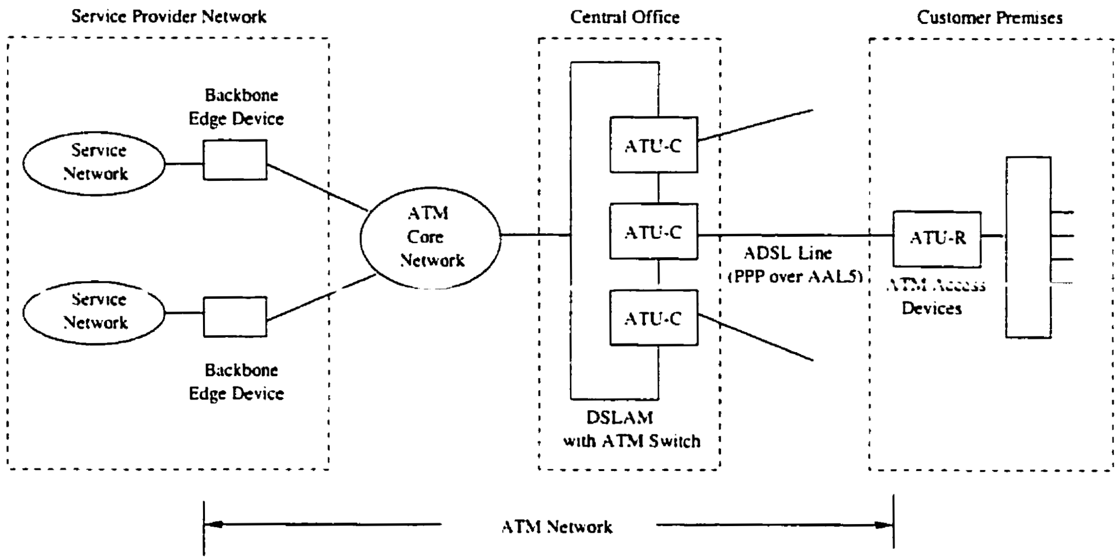


Figure 2.4: TCP/IP over ATM over ADSL Network Architecture [8]

2.3 Outstanding ADSL Issues

ADSL is a technology that can bring broadband services to local loops. However. there are some outstanding issues that still need to be addressed. The challenges were covered briefly in Section 1.2. Here. existing and potential problems related to ADSL are described further.

First of all, noise is an important factor that impairs the performance of ADSL. The transmission medium for ADSL is the traditional copper wire pair, which is subject to many kinds of noise.

Crosstalk is the largest noise contribution, limiting the capacity of ADSL systems. There are two types of crosstalk on multi-pair access network links, Near-End Crosstalk (NEXT) and Far-End Crosstalk (FEXT). NEXT is the interference that appears on another pair at the same end of the link. FEXT is the interference that appears on another pair at the opposite or far end of the link. NEXT affects any systems that transmit in both directions at once, and dominates over FEXT where it exists. NEXT can be eliminated by not transmitting in both directions in the same band at the same time, separating the two directions of transmission either in two non-overlapping time intervals or in frequency bands. This method avoids NEXT at the cost of bandwidth in each direction. Some implementations of ADSL take this approach.

The capacity of ADSL is also limited by line attenuation. Attenuation increases with the increase of line length and frequency. According to the ADSL Forum, a line distance of 18,000 feet can support a downstream rate from 1.5 to 2 Mbps, while a line distance of 9,000 feet can support a speed of up to 6.1 Mbps. Besides crosstalk and attenuation, impulse noise, bridged taps and loading coils are also elements that weaken digital signals.

Another issue under debate is the network architecture based on ADSL. As described in the preceding subsections, the different ADSL transport modes support different network architecture for ADSL. Probably the most common ones are TCP/IP over ADSL and ATM over ADSL.

TCP/IP based servers are everywhere in the world, and many ADSL users are using TCP/IP from the client sites. It is natural to have TCP/IP over ADSL. However, massive changes to existing TCP/IP protocols are required to support QoS for multiple broadband services [14]. An alternative is to use ATM for full service ADSL, which is recommended by the ADSL Forum. ATM is part of the B-ISDN protocol stack and it is designed to support broadband services. The problem with this solution is that ATM implementations are rare on the customer premises.

What needs more study with the TCP/ATM/ADSL architecture is the end-to-end TCP performance. As mentioned earlier in this chapter, TCP suffers performance degradation over ATM because of the fragmentation and reassembly process. TCP also becomes unstable when there exists transmission loss and asymmetry on networks. The network architecture of TCP/ATM/ADSL introduces noise, fragmentation, and asymmetry. The issue is how TCP is affected in terms of end-to-end performance by these factors.

The goal of this study is to explore TCP performance over ATM over ADSL with the existence of network noise and congestion. In this work, a simulation model of TCP/ATM/ADSL was built, network noise was simulated, and the TCP performance under the combination of the factors was studied.

2.4 Summary

This chapter consists of three parts. The first part has briefly surveyed TCP performance under different network circumstances. Research work has shown that TCP performance degrades over ATM networks because of congestion and mismatch of TCP segments and ATM cells. Cell loss due to network noise causes dramatic throughput decrease, especially when consecutive packets are lost. Network asymmetry affects TCP in the way of slowing down the growth of window size and increasing the round time delay. Solutions to mitigate these problems have also been addressed. The second part has reviewed the characteristics and implementation possibilities of ADSL. Emphasis is put on the features that are relevant to this thesis work, including the ADSL components, transport modes, and TCP over ATM over ADSL. In the last part, a discussion has been given about the existing problems of ADSL, such as the noise impact, the network structure, and the performance issues. It has also elaborated why this research was conducted and what was studied.

Chapter 3

The ADSL Simulation Model

A simulation model of ADSL was developed to study the characteristics and performance of TCP over ATM over ADSL (TCP/ATM/ADSL). This chapter describes the design, implementation, and validation of the simulation model. Section 3.1 gives an overview of the ATM simulator, in which the ADSL model is embedded. Section 3.2 discusses the design and implementation of the ADSL simulation model, its structure, and the error models simulating the line noise. Finally, Section 3.3 provides a preliminary validation of the ADSL model.

3.1 The ATM Traffic and Network (ATM-TN) Simulator

The Asynchronous Transfer Mode Traffic and Network (ATM-TN) simulator is an ATM network simulation environment designed and developed by the Telesim research group at the University of Calgary [27]. Background of simulation and

ATM-TN is given in the following subsections because the ADSL simulation model was built upon the ATM-TN simulator in this thesis work.

3.1.1 Discrete-Event Simulation and SimKit

Systems in the real world can be categorized as discrete or continuous. A *discrete* system is one in which the state variables change only at discrete points in time [5]. In contrast, a *continuous* system changes state variables continuously over time. Discrete-Event Simulation (DES) is the approach to model and study discrete systems.

In discrete-event simulation, the system being modeled is usually referred to as the *Physical System*, which contains some *Physical Processes* that interact at various points in time [10]. A simulator is constructed as a set of *Logical Processes (LPs)* , one for each physical process. State transitions of the physical system are modeled by time-stamped messages exchanged between the corresponding LPs. These messages carrying simulation state information are called *events* .

SimKit is a software library that was designed and developed at the University of Calgary for fast discrete-event simulation [12]. The goal of SimKit is to provide an event-oriented logical process modeling interface that facilitates the building of application models for sequential and parallel simulation with high performance execution capabilities.

The SimKit application programming interface (API) includes three basic classes: "Simulation" for simulation control, "LP" for system behavior and state transitions modeling, and "Event" for LP interactions modeling. The user implements an application by deriving application specific classes from LP and Event classes, and by overwriting some methods in the Simulation class.

ATM-TN and the ADSL simulation model in this thesis work are all built using the SimKit API and the discrete-event simulation methodology.

3.1.2 Introduction to ATM-TN

ATM-TN is a high fidelity cell level simulator with a modular architecture that supports the modeling, simulation, and analysis of ATM networks [33]. With ATM-TN, a researcher can define an arbitrary network topology and conduct performance studies under various traffic loads. Simulation was chosen over other methods, such as analytic modeling, because simulation has the capability to model the complex and transient ATM network behavior dynamically. The simulator was designed and developed using the discrete-event simulation methodology. SimKit provides the interface between the network models and the simulation kernel.

ATM-TN consists of three components: network models, traffic models, and a modeling framework.

The network model supports the modeling of arbitrary network topologies. The model is made up of switches, links, end nodes, Virtual Paths (VPs), and Virtual Channels (VCs) [11]. ATM-TN supports three types of switches: the output buffered switch, the shared buffer switch, and the crossbar switch using crosspoint buffering [13]. Each end node is connected to a switch by a single link, and at least one traffic source or sink is associated with one end node. Links are bidirectional. VPs and VCs can be set up along the links. The network model simulates network functionality and behavior including signaling, cell switching, queuing, and congestion control.

The traffic model contains traffic sources and traffic sinks [35]. Traffic sources generate patterns of simulated ATM cells according to the specified traffic types and parameters, and traffic sinks consume incoming cells. The goal of traffic modeling is to design accurate and representative workload models as input for the ATM-TN simulator. Each traffic load is based, to the extent possible, on workload statistics from existing networks, and on the results in the published literature. The traffic models that are supported by ATM-TN include self-similar Ethernet traffic, Motion Pictures Expert Group (MPEG) video streams, Web browsing traffic, generic TCP/IP traffic, and other deterministic and testing models. Each model can be parameterized by the simulator user to represent application specific workload characteristics. The generic TCP/IP model is used in this study. More details of the TCP traffic model will be covered later in Section 4.1.

The modeling framework defines the interfaces to the network and traffic models, the input and output specification, and statistical reporting that are common to many of the sub-models. The framework also contains the simulation control functionality, such as model construction, model initialization, and model execution.

3.2 ADSL Model Design and Rationale

A model is defined as a representation of a physical system for the purpose of studying the system [5]. A model is not a copy of the real system, but an abstraction and simplification of the system. On the other hand, in order to generate valid conclusions, the model should be sufficiently detailed and accurate to reflect the pertinent characteristics of the real system.

Modeling introduces a process of "abstracting" aspects of a real system. The principle of modeling is to abstract the essential features of a system, to select basic assumptions, and to enrich and elaborate the model until useful approximation results.

Building a credible model is an iterative procedure [23]. First of all, a modeler needs to observe the physical system to understand its behavior and characteristics. Most important, the modeler needs to be able to select the essence of the real system that can represent the system and is relevant to the study. It is essential for a modeler to decide what assumptions to make and what should be omitted. A conceptual model is first drawn from the real system, then a computerized model. Measurements of simulation runs can provide observations of the physical system behavior. Comparison of the predicted physical system behavior and the observed behavior from simulation results enables the modeler to estimate the accuracy of the model. Modifications are usually needed to adjust the model to reflect the physical system more closely.

Subsection 3.2.1 describes the major design decisions of the ADSL simulation model and their motivations. The structure of the simulation model is introduced in 3.2.2. The ADSL simulation model includes two error models that simulate the ADSL line noise. They are described in 3.2.3.

3.2.1 Model Design

The simulation model design was motivated mainly by four objectives.

- Focused. It is not necessary for the simulation model to represent all features of ADSL. In this study, emphasis is put on the effect of noise on the end-to-end network performance.

- Generalized. Although the thesis project is focused on a study of ADSL, it need not be limited to ADSL. The features that are simulated and studied, such as noise and network asymmetry, can also exist on other networks, for example, wireless and broadcast satellite networks. The goal of the thesis is to study some network phenomena in the context of ADSL, but not confined to ADSL. The model is designed to be flexible so that different network characteristics can be defined by different parameters. Therefore, the system can also be configured to model networks with similar characteristics.

- Embedded. The ADSL simulation model is built upon the existing ATM-TN simulator. This makes it possible to utilize the network components and protocols that have been simulated in ATM-TN, such as the ATM switch, traffic sources, ATM, and TCP/IP protocols. The model is designed to be able to fit within ATM-TN, while ATM-TN remains unchanged.

- Extensible. The simulation model is modularized and reusable. Changing or adding features can be accomplished with minimal effort.

3.2.2 The ADSL Simulation Model

The ADSL simulation model is built upon the existing ATM-TN simulator. It utilizes the functionality of ATM-TN and the TCP traffic module in the ATM-TN. In order to simulate specific ADSL features, it extends the ATM-TN simulator with ADSL components and introduces noise over ADSL lines. This subsection and the following subsection describe the design and development made in this thesis to simulate the ADSL Access Node, ADSL modems, and noisy ADSL lines.

The ADSL simulation model is based on the ADSL Forum recommended ATM over ADSL model. Figure 3.1 illustrates how the simulation model is abstracted from the ATM/ADSL reference model defined by the ADSL Forum. The upper diagram is the simplified ADSL Forum recommended model [1]. The lower diagram is the simulation model. The arrows in between show how the key components of the ADSL Forum reference model are matched by the simulation model.

The key ADSL components in a real system include the Access Node in the central office, ADSL Transmission Unit-Remote Terminal (ATU-R) at customer premises, and ADSL lines in local loops. Access Node consists of the ATM access switch, Digital Subscriber Line Access Multiplexer (DSLAM), and ADSL Transmission Unit-Central Office (ATU-C). In the simulation model, there are three structures to match their counterparts in the real system. They are the ADSL Access Node, ADSL Remote devices, and ADSL lines. The ADSL Access Node includes an output buffered single stage ATM switch, the multiplexer and demultiplexer, and ADSL central office devices. Both the ADSL Central and ADSL Remote devices have the functionality of error detection and error handling. ADSL lines are error-prone connections. Error characteristics for each line can be defined by simulation parameters. Although there are different types of noise over the transmission media, their effects are the same, i.e., errors on the ADSL lines. The only difference is that different types of noise may cause different loss patterns. To simulate the noise at the physical layer, two error models have been implemented. The details of the error models will be covered by Subsection 3.2.3.

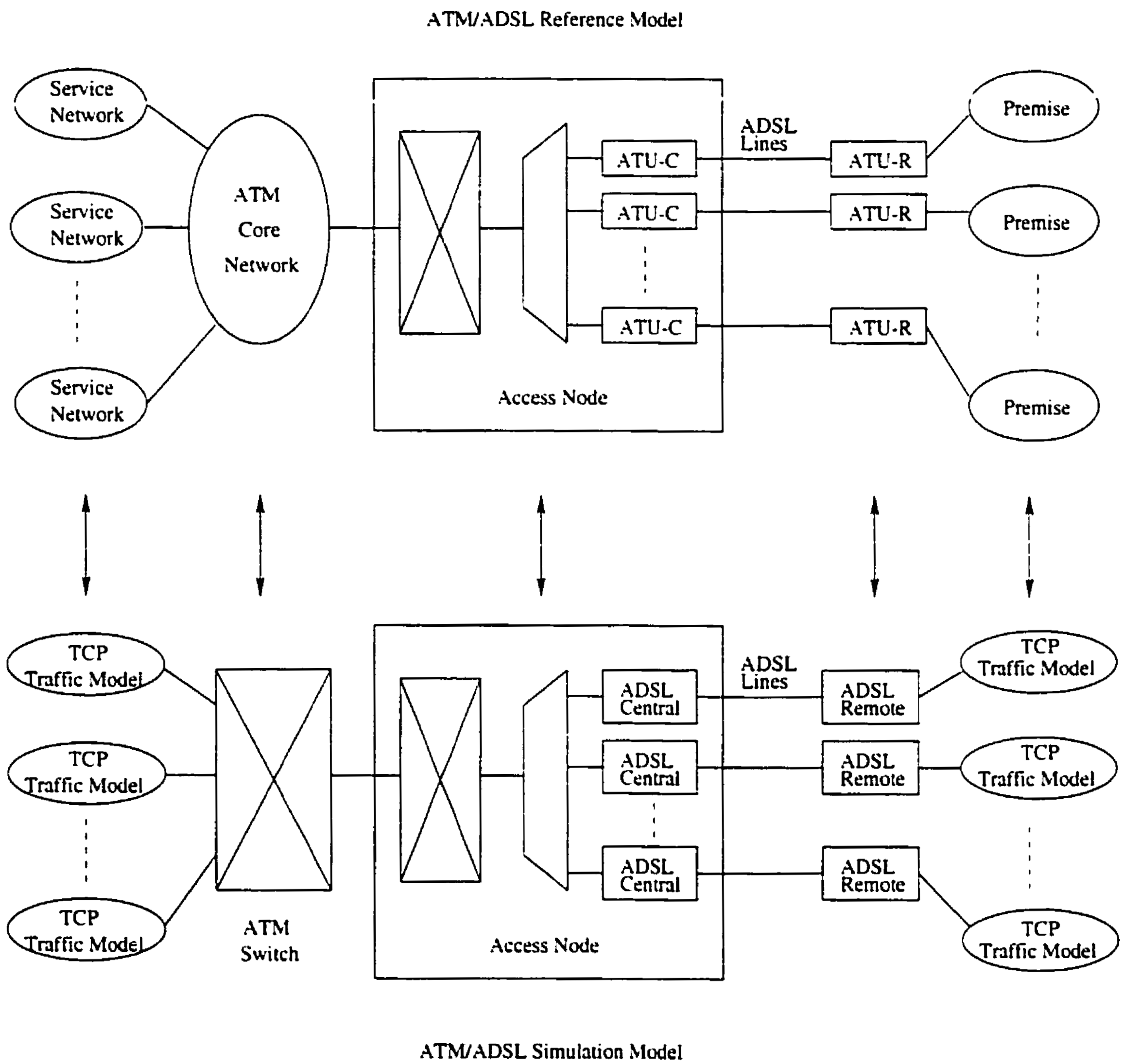


Figure 3.1: Comparison of the ATM/ADSL Reference Model and the Simulation Model

Figure 3.2 shows the network structure when the ADSL simulation model is used on the access network. The ADSL Access Node, ADSL Remote Devices, ADSL lines, and TCP clients form the ADSL access network. In the diagram, downstream refers to the traffic streams from service networks to customer premise networks. Traffic is demultiplexed by the DSLAM and sent to different customers. Upstream traffic flows in the reverse direction. Traffic streams from multiple ADSL lines are multiplexed by DSLAM and switched by the ATM access switch at the ADSL Access Node, sent to the ATM switch in the core ATM network, and transmitted to different service networks. For downstream traffic, cells may be corrupted when they traverse the noisy ADSL lines. These errors are detected by the corresponding ADSL Remote Device and errored cells are dropped before they reach the client site. Similarly, cells in the upstream traffic may be corrupted when they are transmitting on the ADSL lines from the ADSL Remote Devices to ADSL Central Devices. These errored cells are detected and dropped by the ADSL Central Devices before they enter the core ATM network.

3.2.3 Error Models for ADSL Line Noise

An important feature of ADSL that is studied is the random errors due to noise over copper lines. This subsection describes the two simulation error models that have been implemented to characterize the features of noise on ADSL lines. Since ATM-TN is a cell level simulator and the smallest unit in ATM-TN is the ATM cell, the error models simulate errors at the ATM cell level, not at the bit level.

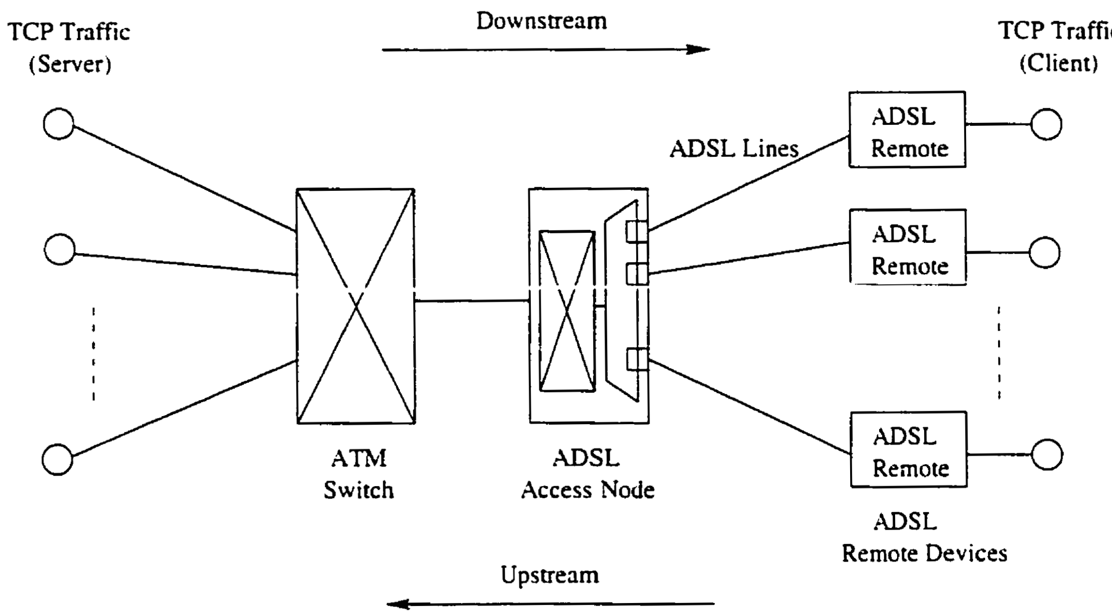


Figure 3.2: An ADSL Network Based on the Simulation Model

Independent Error Model

In the independent error model. the probability of a cell being corrupted is independent of that of other cells, so that errors are scattered. There are two parameters in the simulation input file to be used by this error model: the Cell Error Ratio (CER) in the downstream and upstream directions. CER is the ratio of errored cells in a transmission in relation to the total cells sent in the transmission. These two parameters define the probability that cells get corrupted and dropped.

For each incoming cell, a random draw is made to generate a random number between (0-1) using the SimKit random number generator. If the number is less than or equal to the probability given as the parameter (e.g. CER $= 1.0 \times 10^{-7}$), it indicates an error and the cell (and only this cell) is dropped. Otherwise, no drop occurs and the cell is sent to the destination.

Burst Error Model

Burst error is the most common error pattern in real systems. A burst error can involve consecutive cells or packets of a data stream. The simulation of burst errors can be in the time domain or in the cell count domain. The time domain approach defines the line status in terms of time, for example, noise occurs from time t_1 to t_2 on a line. All the cells (zero or more) transmitted over the line during this noisy period will be corrupted. The cell count approach defines the line status in terms of the number of corrupted cells in an error burst, for example, a mean of n cells will be lost in an error burst. The time domain simulation approach is used in this study. There are four parameters in the simulation input file for the burst error model: CER in the downstream direction, CER in the upstream direction, a mean burst error duration in seconds in the downstream direction, and a mean burst error duration in seconds in the upstream direction.

The key issue in the burst error model is how to calculate the interarrival time between the bursts of errors. A cell stream with burst errors is illustrated in Figure 3.3.

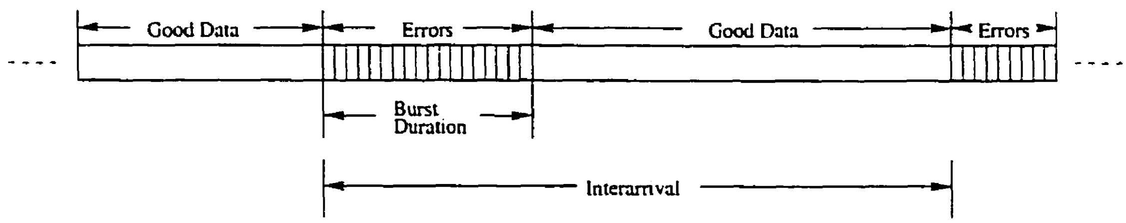


Figure 3.3: The Burst Error Pattern

The following approach is used. Select any period with a section of good data and a section of errors. Then the following proportion expression can be derived:

$$\frac{ErrorCells}{ErrorCells + GoodCells} = \frac{Line\_Rate \times BurstDuration}{Line\_Rate \times Interarrival} \tag{3.1}$$

ErrorCells refers to the number of corrupted cells during the selected period, marked as *Errors* in Figure 3.3. *GoodCells* refers to the number of correctly transmitted cells, marked as *Good Data* in Figure 3.3. Since the left part of the equation is equal to CER according to the definition of CER (the ratio of errored cells in a transmission to the total cells sent in the transmission), the equation can be written as:

$$CER = \frac{BurstDuration}{Interarrival} \tag{3.2}$$

Therefore, the interarrival time can be calculated as:

$$Interarrival = \frac{BurstDuration}{CER} \tag{3.3}$$

Both the $BurstDuration$ and CER are given in the input data set as parameters so that the interarrival time can be directly calculated.

Since the bursts of errors follow the Poisson process, the exponential distribution is used. One exponential distribution is used to determine when the burst of errors occurs, and another exponential distribution is used to determine how long the burst error lasts (i.e., burst error duration). The mean for the former is the "interarrival" time calculated as above. The mean for the latter is the "mean burst error duration" in seconds given in the simulation input data set. The status of an ADSL line changes between "good" and "noisy". If a cell arrives during the noisy period, the cell is considered corrupted and dropped. Otherwise the cell is sent to the destination.

3.3 Model Verification and Validation

Model verification assures that the computer simulation program reflects the conceptual model accurately. Validation refers to the act of determining that the simulation model represents the real system well enough for the purposes of the study [5]. Since verification and validation are usually conducted simultaneously, they will not be strictly distinguished in this section.

Verification and validation are important and difficult tasks for model developers. Many methods introduced in the literature are informal subjective comparisons and judgments, while a few are based on the statistical analysis of the output. This section provides the preliminary model verification and validation by conducting three groups of experiments. These experiments are: 1) an illustration of the behavior of the two error models, 2) comparison of the expected results and simulation results, and 3) simulation experiments with different random number seeds. Experiments have been run with a single TCP source scenario for model validation. Please refer to Section 4.2 for the details of the scenario configuration.

3.3.1 Error Model Behavior

Figure 3.4 provides a qualitative verification of the error models by plotting the error distributions as a function of time. For the burst error model, the vertical axis is the number of cells dropped in an error burst. Since the mean duration of each error burst is very short ($200\mu s$) compared with the simulation time in the horizontal axis, the duration of an error burst cannot be seen from the graphs. Instead, an error burst is shown as an impulse in the graphs. The number of cells corrupted in an error burst varies from one to five. The error distribution is as expected, i.e., the lower the Cell Error Ratio (CER), the sparser the burst errors. The plot for CER = 1.0×10^{-4} shows fewer corrupt cells in an error burst although the same mean burst error duration is used for the two CERs. It is because the error bursts are more scattered when CER equals 1.0×10^{-4}. With a longer simulation time, the average number of cells corrupted in an error burst for these two CER values becomes closer. As it was designed, the errors in the independent model are not correlated to each other so that only one cell is corrupted each time.

The independence of random numbers is validated by performing a test for auto-correlation according to the method introduced in [5]. In testing for independence, a null hypothesis is made that the numbers are independent. Failure to reject the null hypothesis means that no evidence of dependence has been detected on the basis of the test.

The test is conducted as follows [5]: 1) Select a sequence of random numbers for study, 2) Examine every m numbers, starting with the i_{th} number. Thus the numbers R_i, R_{i+m}, R_{i+2m}, $\ldots$, $R_{i+(M+1)m}$ would be of interest. The value M is the largest integer such that $i + (M + 1)m \leq N$, where N is the total number of values in the sequence, 3) The test statistic is formed as $Z_0 = \hat{\rho_{im}} \,/\, \sigma_{\hat{\rho_{im}}}$. For large values of M, the distribution of $\hat{\rho_{im}}$ is approximately normal with a mean of zero and a variance of 1 if the random numbers studied are uncorrelated. The formulas for $\hat{\rho_{im}}$ and $\sigma_{\hat{\rho_{im}}}$ are given in Equation 3.4 and 3.5. 4) Compute Z_0, and do not reject the null hypothesis of independence if $-z_{\alpha/2} \leq Z_0 \leq z_{\alpha/2}$, where α is the level of significance and $z_{\alpha/2}$ is the critical value.

In the thesis, a sequence of 30 random numbers are selected for the study. α is 0.05. Multiple computations are conducted with different i and m. This increases the probability of rejecting the null hypothesis. All the statistical values of Z_0 meet the criteria in step 4). Therefore, the hypothesis of independence cannot be rejected by the test results.

$$\hat{\rho_{im}} = \frac{1}{(M+1)}[\sum_{k=0}^{M} R_{i+km}R_{i+(k+1)m}] - 0.25 \tag{3.4}$$

$$\sigma_{\hat{\rho_{im}}} = \frac{\sqrt{13M+7}}{12(M+1)} \tag{3.5}$$

Figure 3.5 compares the error distributions with the same CER but different mean burst error durations. The graphs illustrate that with a longer mean burst error duration, more cells are corrupted in an error burst, but the interarrival time between error bursts tends to be larger. Therefore, the CERs are the same according to Equation 3.2.

Figure 3.6(a) shows the cumulative number of dropped cells and its relationship

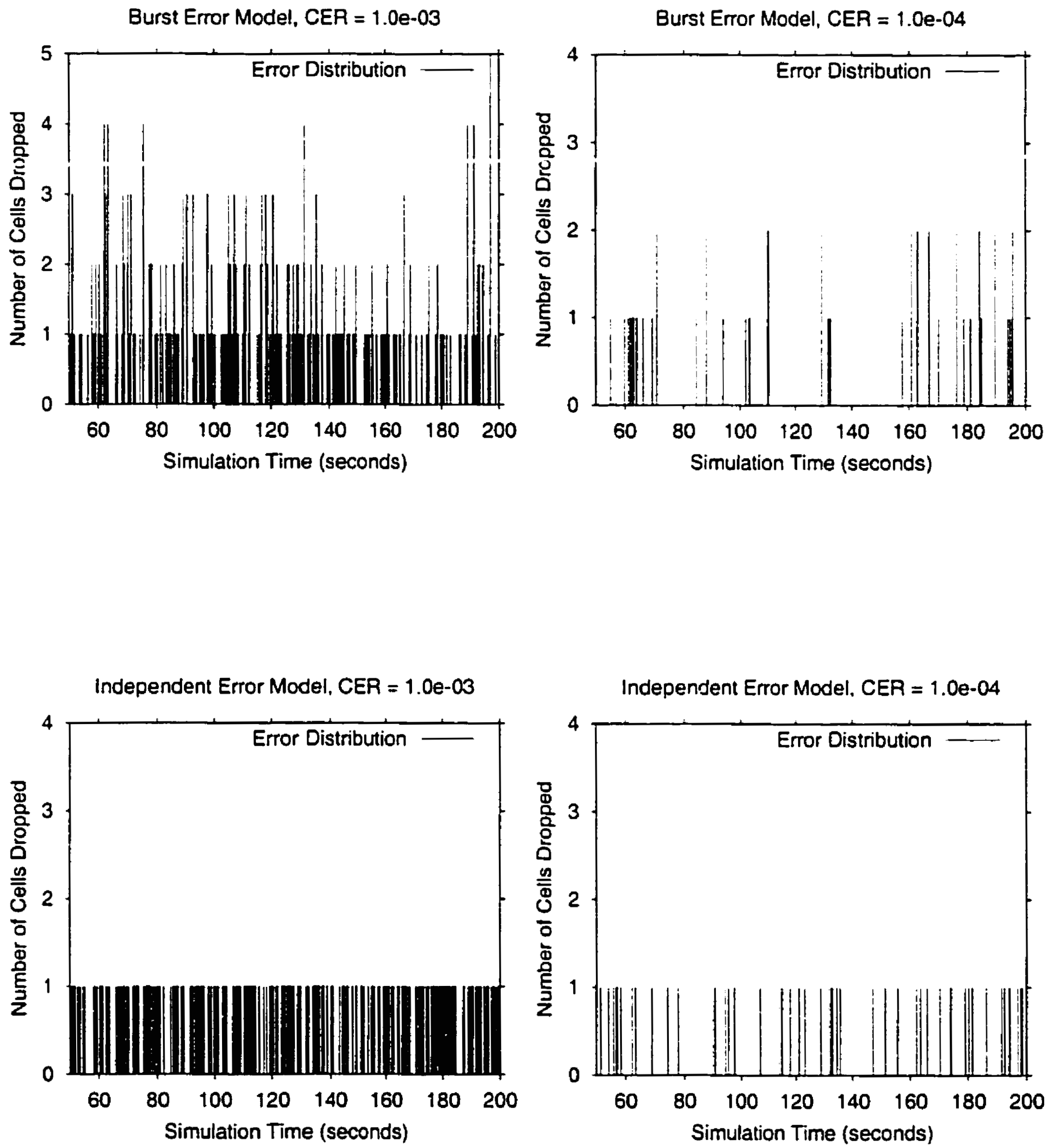


Figure 3.4: Comparison of Error Distributions of the Burst and Independent Error Models

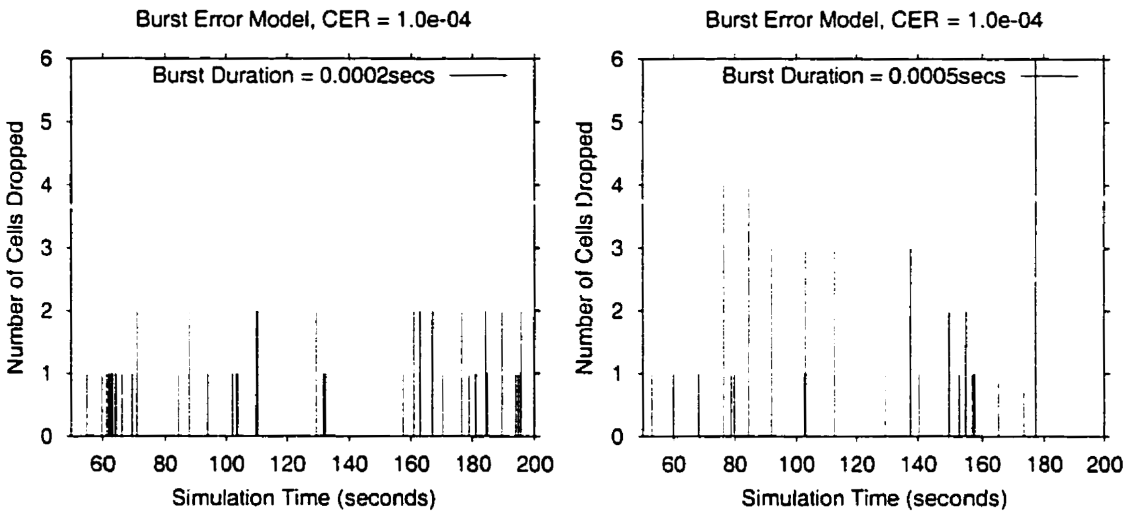


Figure 3.5: Comparison of Bursts of Errors with Different Burst Durations

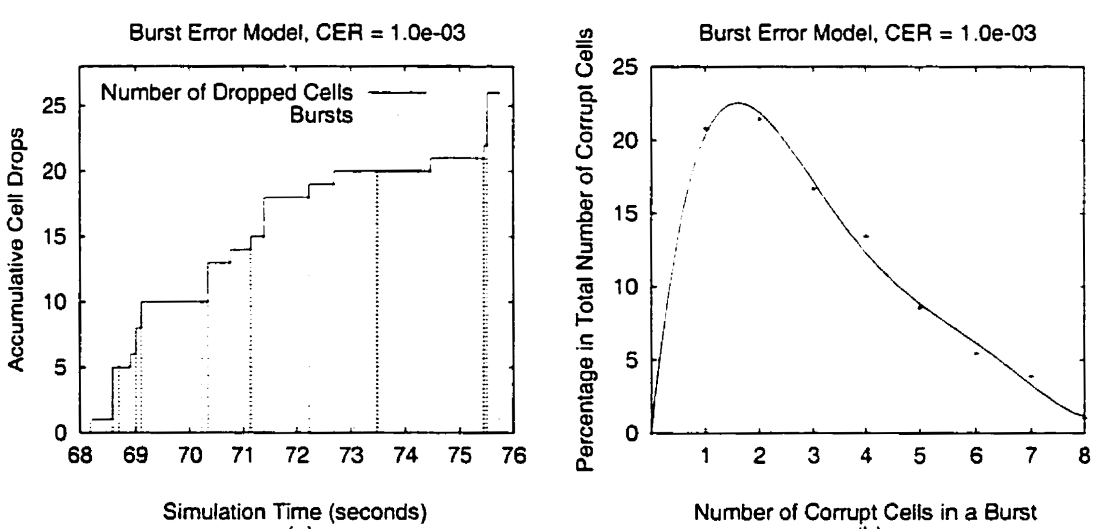


Figure 3.6: (a) Cell Drops Accumulation as a Function of Time (b) Distribution of Number of Cells Corrupted in a Burst

with error bursts within a short simulation period. The error bursts are illustrated as impulses with dotted vertical lines. It is seen that each increment of the cumulative number of dropped cells is caused by an error burst. Some error bursts cause no change in the cumulative number of dropped cells because there is no cell transmission during the error burst.

Figure 3.6(b) illustrates the distribution of the number of cells corrupted in a burst error. The horizontal axis shows that a burst error may corrupt a single cell or up to eight consecutive cells. The vertical axis shows their percentages in total number of corrupted cells respectively. For example, the number of cells corrupted in error bursts that kill one cell accounts for 20.8% in the total number of corrupted cells. The curve is a polynomial approximation of the distributions. The average number of cells corrupted in an error burst is 2.22 as shown by the dotted vertical line. Error bursts that destroy one or two cells are dominant because the interarrival time of cells is very close to the mean burst error duration in the experiments.

3.3.2 Comparison of Targeted and Observed Results

This subsection performs the model validation by comparing the targeted and observed results. Targeted results are expected results from the simulation definition. Observed results are obtained from the simulation statistics.

Table 3.1 and 3.2 compare the targeted and observed data for the error models. Targeted CER is defined in the simulation input file. Observed CER is calculated as the ratio of dropped cells to received cells. It is expected that the observed CER is the same as the targeted CER. The simulation results have a good match to the targeted data, which indicates that the models work as they were designed.

Table 3.1: Comparison of Targeted and Observed Data (Burst Error Model)

Targeted CER	Observed Data		
	Cells Dropped	Cells Received	Observed CER
1.0×10^{-3}	149,415	149,459,669	1.00×10^{-3}
1.0×10^{-4}	17,463	173,855,383	1.00×10^{-4}
1.0×10^{-5}	1,826	176,186,389	1.03×10^{-5}
1.0×10^{-6}	166	176,646,027	0.94×10^{-6}
1.0×10^{-7}	21	176,709,724	1.19×10^{-7}

Table 3.2: Comparison of Targeted and Observed Data (Independent Error Model)

Targeted CER	Observed Data		
	Cells Dropped	Cells Received	Observed CER
1.0×10^{-3}	141,341	140,780,961	1.00×10^{-3}
1.0×10^{-4}	17,353	174,140,195	1.00×10^{-4}
1.0×10^{-5}	1,805	176,582,676	1.02×10^{-5}
1.0×10^{-6}	188	176,708,014	1.06×10^{-6}
1.0×10^{-7}	17	176,709,736	0.96×10^{-7}

3.3.3 Experiments with Different Random Number Seeds

The simulation model should produce similar results with different sequences of random numbers. Three different random number seeds have been used to generate three different sequences of random numbers to test the stability of the simulation model. Figure 3.7 shows the results of targeted CER for the two error models with different random number sequences. Logarithmic scales are used for both the horizontal and vertical axes in order to illustrate the differences clearly. It is seen that different random number sequences generated very close results. This proves that the simulation model is not subject to the change of random numbers.

Figure 3.7 also visualizes the comparison of targeted and observed data in the preceding subsection. Since the observed CER should be the same as the targeted CER, the ideal line in the graphs should be the diagonal. The close approach of the lines to the diagonal indicates that the model performed as it was designed.

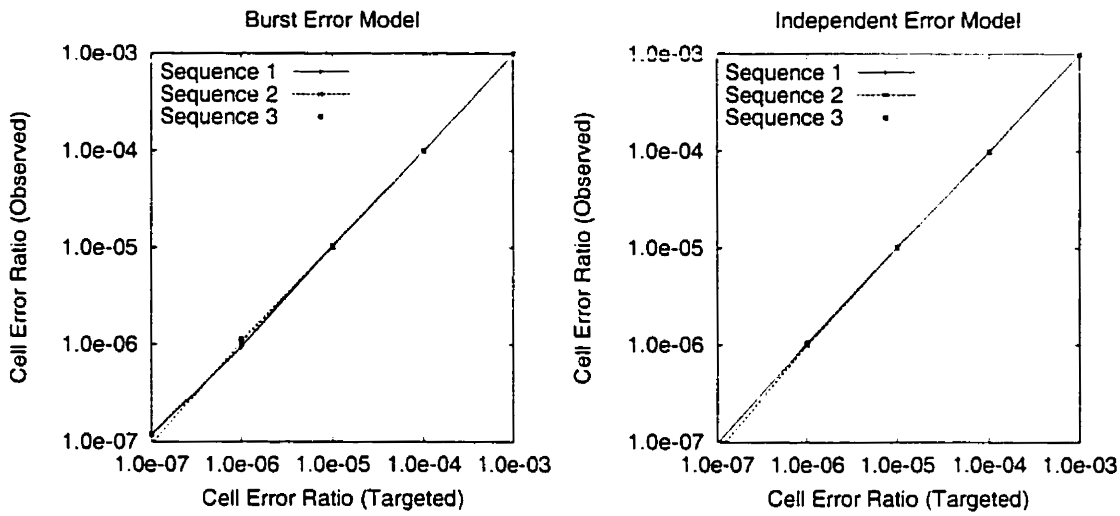


Figure 3.7: Comparison of Results with Different Random Number Seeds

3.4 Summary

This chapter has described the design, implementation and validation of the ADSL simulation model. The model is built upon a high fidelity ATM network simulator called the ATM-TN. It enables the study of a full protocol stack of TCP/ATM/ ADSL. Major ADSL components have been simulated. Noise on ADSL lines is simulated by two error models and is configurable via simulation parameters. A set of experiments have been performed for model verification and validation. Simulation results have shown that the model works as it was designed.

Chapter 4

Experimental Design

The main objective of the thesis is to explore the effects of a noisy ADSL access network on the end-to-end TCP performance over ATM (denoted as TCP/ATM/ADSL). This is achieved by conducting a series of experiments and analyzing the experimental results. The method of doing the study is simulation. ATM-TN is an integrated simulation tool that enables the users to construct the TCP/ATM/ADSL structure and to define different network scenarios.

This chapter describes the design of simulation experiments in the thesis work. Section 4.1 introduces the TCP traffic simulation model that is used in the experiments. Section 4.2 describes the network structure for the experiments. Section 4.3 presents the parameters and their values that affect the simulation results. Section 4.4 defines the performance metrics used in the result evaluation. Section 4.5 describes the purposes and designs for each group of experiments.

4.1 The TCP Traffic Model

The TCP model in ATM-TN was designed and implemented to simulate the TCP/IP protocol suite and generate the traffic load. The TCP model is closely based on the TCP/IP networking code from the 4.4BSD-Lite release by the University of California at Berkeley, which is essentially the implementation of the TCP-Reno version. The model is designed to simulate the data transfer between two hosts over an ATM network. Both hosts can send and receive data although they are referred to as the source (the primary sender) and the sink.

Figure 4.1 depicts the protocol stack of the TCP traffic model. There are four sections in the stack: the application level, the socket interface, the TCP/IP protocols, and the AAL5/ATM layer [15]. In the application layer, each host writes a certain amount of data to the destination. The data is buffered by the socket layer. The TCP layer is modeled in great detail including TCP features such as slow-start, fast retransmit, fast recovery, and high performance extensions. The IP layer is not explicitly modeled. The routing is left to the ATM layer. The AAL5/ATM layer breaks TCP segments into ATM cells and queues them for transmission. The received cells are reassembled into TCP segments, checked for completeness, and passed to the TCP/IP layer.

The TCP model was implemented as a traffic module in ATM-TN using the SimKit C++ version [12]. It can be configured by parameters in the ATM-TN input data sets. These parameters include the finite or infinite amount of data to transfer, the socket send and receive buffer size, TCP timer granularity, Maximum Segment Size (MSS), Maximum Transmission Unit (MTU), TCP high performance extensions, and so on. The TCP model also provides the reporting and tracing

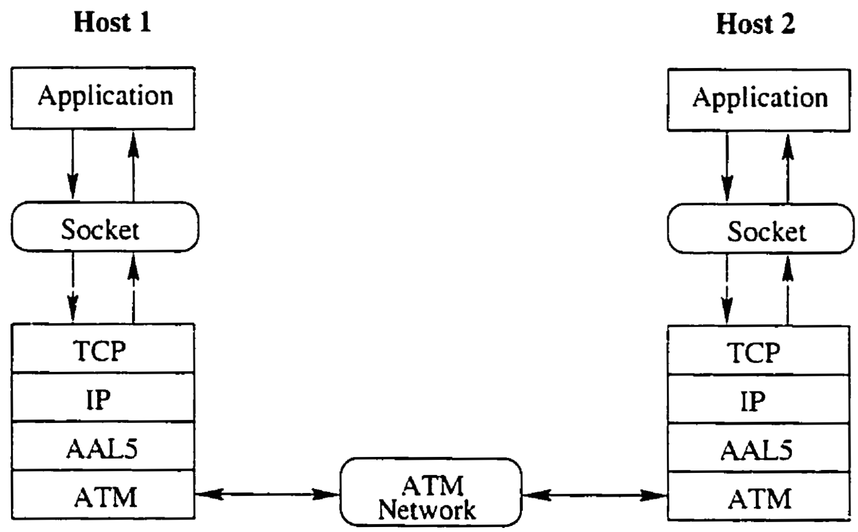


Figure 4.1: The TCP Traffic Model [15]

functions for post simulation study.

4.2 Network Scenarios

Two network scenarios have been used in the simulation experiments. Figure 4.2 shows the structure of the single TCP source scenario. It is used to explore the dynamic behavior of TCP. In order to be consistent with the notation in the literature, the customer premise network is shown on the right side of the graph, and the service network is on the left side. In the simulation experiments, the TCP traffic instances are configured to be unidirectional, i.e., the server will send unlimited data packets to the client, and the client will only send acknowledgments to the server. This simulates the typical asymmetric operation on the Internet, where users usually download more data than they upload. The ATM service class is Unspecified Bit Rate (UBR) in all the experiments.

There are two kinds of links in the scenario: the high-speed ATM links and low-speed ADSL lines. ATM link rates are defined as 155 Mbps (OC-3). The downstream transmission rate for the ADSL line is 1.5 Mbps. The upstream rate for ADSL is 512 Kbps. The ADSL line is the bottleneck of the network because of the rate mismatch. ATM links and the ADSL line are distinguished using thick and thin lines respectively in the diagram. Link distances in kilometers are marked beside the lines. The ADSL line is 3.5 Km, a typical length in local loops. The link from the TCP client to the ADSL remote device is negligible because it is an internal link. In the simulation it is defined as a very short error-free link with the same transmission rate as the ADSL line.

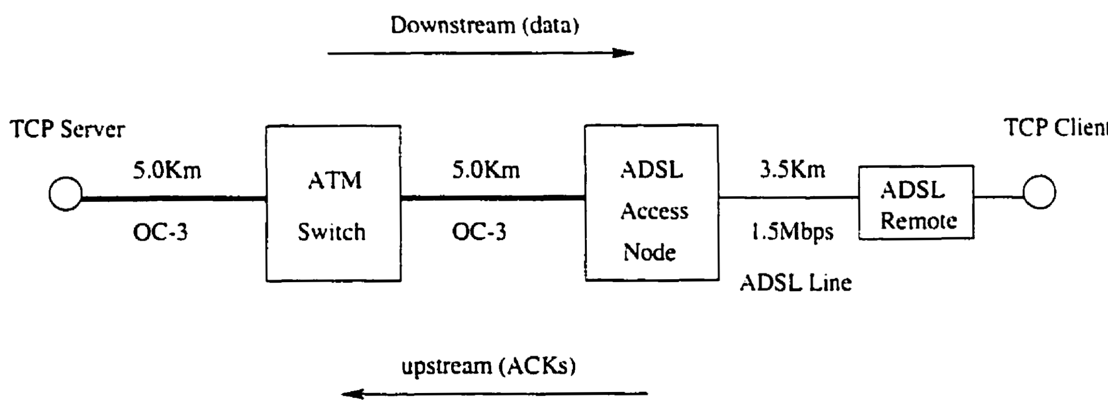


Figure 4.2: The Single TCP Source Scenario

The second scenario has similar structure but eight TCP sources as shown in Figure 4.3. It is used in most of the experiments. In this scenario. eight TCP sources are multiplexed by the ADSL Access Node. Eight Permanent Virtual Connections (PVCs) are established between the clients and servers. The length between the ATM switch and ADSL Access Node is 5 Km for Local Area Network (LAN) and

1000 Km for Wide Area Network (WAN). All the experiments are based on the LAN configuration unless otherwise specified. Other parameters for the eight TCP sources scenario are the same as the single TCP source scenario.

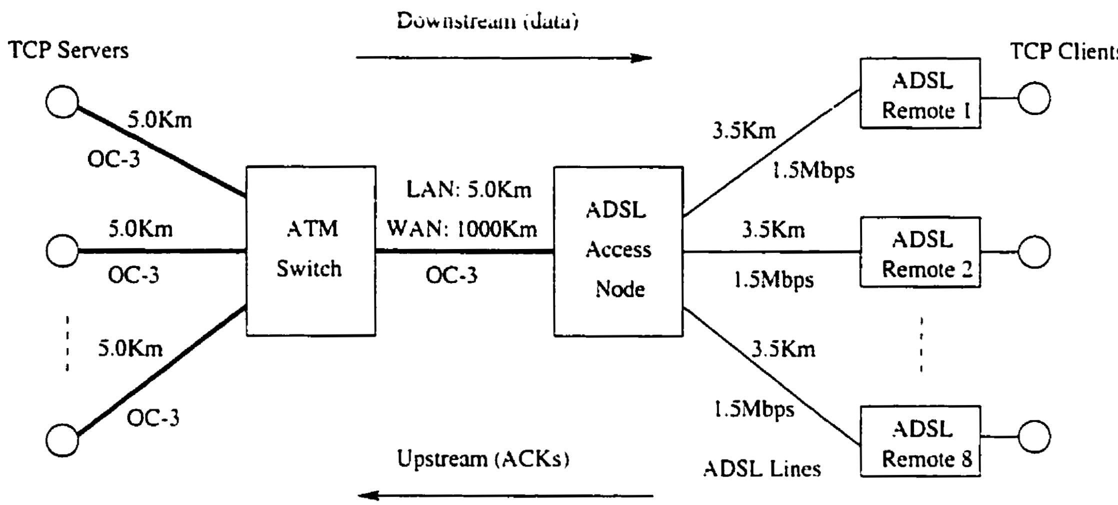


Figure 4.3: The Eight TCP Sources Scenario

4.3 Factors and Non-factor Parameters

Factors are the parameters that influence the experimental results and are varied with different values in the experiments. Non-factor parameters define the network scenario and are set to fixed values in the experiments.

4.3.1 Factors and Levels

The factors used in the experiments are described below. Their levels are listed in Table 4.1.

- Cell Error Ratio (CER): According to the ATM Forum, Cell Error Ratio is defined as the ratio of errored cells in a transmission in relation to the total cells sent in the transmission. For example, CER $= 1.0 \times 10^{-7}$ means that an average 1 out of every 10,000,000 cells is corrupted. Although errors normally appear in bursts, carriers usually use CER as a descriptor of the transmission quality ignoring the non-uniform distribution of errors. In the experiments, the value of CER is set from 1.0×10^{-3} to 1.0×10^{-7}. The CER values for the downstream and upstream directions of the same ADSL line are the same.

- Burst Error Duration: Burst error duration defines the mean length of bursts of errors in seconds. The frequency of the occurrence of errors and the duration of bursts determine the CER of a line. The duration of noise in a real system varies according to different types of noise. For example, the duration of impulse noise may range from a few μs to 1000 μs [6]. In most of the experiments, the burst error duration is set to 200 μs. In the experiments considering the effect of noise (Section 5.2), mean durations of 80 μs and 500 μs are also studied.

- Maximum Segment Size (MSS): Maximum Segment Size defines the maximum amount of data in bytes in a TCP segment (TCP packet). In most experiments, the MSS is set to 512 bytes, the packet size used commonly on the Internet. In the experiment to study the effect of MSS, two other values are studied. One is 4352 bytes for FDDI networks. The other is 9140 bytes, the default MSS size for IP over ATM networks.

- ADSL Line Bandwidth: ADSL line bandwidth is the transmission rate of

ADSL lines in Megabits per second (Mbps). In most of the experiments, the ADSL line bandwidth is fixed at 1.5 Mbps except that in the bandwidth asymmetry study it is varied from 1.5 Mbps to 8.0 Mbps. All the ADSL lines are defined to have the same bandwidth.

- Percentage of noisy ADSL lines within a certain network: This is a unique parameter introduced in this thesis. It is defined as the ratio of noisy ADSL lines to the total number of ADSL lines on a network. Given a network with a total of 1000 ADSL lines, if 100 lines cause transmission errors due to noise, while other lines are in good status, the percentage of noisy lines is said to be 10%.

- Switch Buffer Size: Switch buffer size is the output port buffer capacity of the ATM switch in cells. When congestion occurs, cells are queued in the buffers. When the buffer overflows, incoming cells are dropped. In most experiments, the buffer size is set large enough to avoid cell drops by congestion, thus isolating the cause of data loss to errors. In the experiment studying the effect of switch buffer size on network performance, the buffer size varies and there are cell drops due to buffer overflows.

4.3.2 Non-factor Parameters

- Link Features: Two types of links are defined in the experiments: ADSL lines and ATM links. ADSL lines refer to the low-speed copper pairs in local loops. ATM links refer to high-speed fiber optic links. For the ADSL lines, the upstream transmission rate is set to 512 Kbps and the downstream rate

Table 4.1: Levels of Factors

Factors	Levels
Cell Error Ratio (CER)	1.0×10^{-3}, 1.0×10^{-4}, 1.0×10^{-5}, 1.0×10^{-6}, 1.0×10^{-7}
Burst Error Duration (seconds)	0.00008, 0.0002, 0.0005
Maximum Segment Size (Bytes)	512, 4352, 9140
ADSL Line Bandwidth (Mbps)	1.5, 2.5, 4.0, 6.0, 8.0
Percentage of Noisy Lines	0%, 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, 100%
Switch Buffer Size (Cells)	1000, 1100, 1200, 1300, 1400, 1500, 1600

is 1.5 Mbps. The propagation delay of ADSL lines is 5.7 μs per kilometer. ATM links are defined as OC-3 links with a bandwidth of 155.0 Mbps in both directions and a delay of 3.7 μs per kilometer. The ATM links are error-free in the simulation.

- Switch Features: The switch type selected in the experiments is the single stage Per-Port switch model in ATM-TN [11]. When congestion occurs, cells are queued in the output port buffers of the switch.

- TCP Features: In the experiments, the TCP model is configured as one-way traffic, i.e., unlimited data is sent downstream and only acknowledgments are sent upstream. This simulates a user's behavior of sending requests to a network and downloading a large amount of data from the network. The TCP receiver's window size is 65,535 bytes. A TCP connection's capacity is decided by bandwidth-delay product (bandwidth $\times$ round-trip time). If the window size is greater than the bandwidth-delay product, the maximum possible throughput can be achieved. 65,535 is large enough not to be a restriction on the throughput. The Maximum Segment Size (MSS) is set to 512 bytes except for the study of different MSS values. The TCP retransmission timer granularity is 500 ms because it is commonly used on the Internet. Delayed acknowledgment is used (200 ms delay). Nagle's algorithm is enabled, and TCP high performance extensions are disabled.

- Simulation set up: In order to obtain steady-state results, the simulation run time and warmup period need to be long enough. In this thesis, the simulation is run for 5000 simulation seconds with a warmup time of 50 seconds. These values were chosen by running tests with different simulation parameters. In the test runs, the simulation trace function is turned on so that it generates simulation statistics at the configured intervals. By examining the statistics, for example, the change of maximum and average queue size, it can be determined whether or not the simulation program has reached the steady state. In the real experiments, the trace function is turned off so that the statistics are collected after the warmup period and generated once at the end of the simulation. The default random number generation mixer of ATM-TN is used. As seen in Subsection 3.3.3, simulation with different random number seeds generated similar results.

4.4 Performance Metrics

The metrics for the evaluation of the performance of TCP/ATM/ADSL are explained as follows:

- Cell Loss Ratio (CLR): CLR is the ratio of the number of lost cells to the total number of cells sent over a network within a certain time interval [26]. It is an ATM specific metric for the measurement of quality of service on a network. It is expressed as an order of magnitude, usually having a range from 10^{-1} to 10^{-15}.

- Packet Loss Ratio (PLR): PLR is defined as the ratio of number of lost packets to the total number of transmitted packets. PLR is the metric to study data loss at the TCP/IP layer.

- Effective Throughput: The effective throughput is defined as the ratio of the aggregate *achieved* throughput to the maximum *achievable* throughput. It is calculated by the following equation.

$$EffectiveThroughput = \frac{\sum_{i=1}^{n}(Throughput)}{N \times LineRate \times \frac{MSS}{53 \times \lceil (MSS + 40) \div 48 \rceil}} \tag{4.1}$$

$\sum_{i=1}^{n}(Throughput)$ is the aggregate *achieved* throughput. It is the amount of data successfully transmitted (excluding duplicate data) divided by the transmission time, usually expressed as Mbits/sec. It can be accumulated from the simulation statistics. The denominator is the maximum *achievable* throughput of the network. It is limited by the bottleneck on the network. LineRate (Mbits/sec) is the maximum transmission rate of the bottleneck link, which is the ADSL line in this study. N is the number of bottleneck links on a network. In the network scenario shown in Figure 4.3, N is 8.

$\lceil (MSS + 40) \div 48 \rceil$ calculates the number of cells needed to carry a TCP segment. There is overhead to pad TCP segments into ATM cells because each ATM cell (53 bytes) contains only 48 bytes of payload. For example, a TCP segment of 512 bytes of data (MSS = 512 bytes), plus 20 bytes TCP header and 20 bytes IP header, needs 12 ATM cells (636 bytes). So the *achievable* throughput is 80.5% (512 bytes/636 bytes) of the bottleneck bandwidth. An effective throughput of 100% is desired, which means the full utilization of the link bandwidth.

- Fairness Index: Fairness means the equal allocation of resources, for example, equal share of the bottleneck link on a network. It is desirable for a network to offer its resources fairly to contending connections. With an unfair allocation, some connections get better performance at the expense of other connections even though the total throughput is high. The fairness index is a measure of fairness as introduced in [20]. It is expressed by the following equation:

$$FairnessIndex = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{4.2}$$

  In the equation, n is the number of contending users and x_i is the resource allocated to the i_{th} user. The value of the fairness index is continuous and bounded between 0 and 1, where 1 means a system that is 100% fair.

- TCP Congestion Window: The congestion window is a state variable used by the TCP slow start algorithm. Upon starting a connection, or restarting from slow start after packet loss, the congestion window size is set to one packet. The sender can transmit a number of packets that is up to the minimum of the congestion window size and the advertised window size [30]. The congestion window when plotted against time is used to illustrate and explain the dynamic behavior of TCP.

- TCP Sequence Number: Each octet (byte) of data sent over a TCP connection is logically assigned a 32-bit sequence number [18]. The acknowledgment mechanism is cumulative so that an acknowledgment of sequence number n indicates that all octets up to but not including n have been received. In other words, the acknowledgment number indicated in the ACK is the next sequence number expected by the receiver. Since every octet is sequenced, each can be acknowledged. This mechanism supports the reliable data transmission of TCP. The plot of sequence numbers also helps to observe and analyze the dynamics of TCP.

4.5 Experimental Design

Several groups of experiments have been designed to explore the performance of TCP/ATM/ADSL under a lossy environment.

- Dynamics of TCP: This experiment explores the dynamic behavior of TCP with the existence of errors by plotting the TCP congestion window and sequence numbers against time. These two variables provide a close view of TCP's reaction to data loss. The acknowledgment of duplicate sequence numbers indicates TCP packet loss. The congestion window size is reduced when packet loss is detected. The experiment is conducted with the single TCP source scenario in a short simulation time in order to observe changes in these variables.

- The effect of noise: The purpose of this experiment is to study the effect of noise in isolation. Cell Error Ratio (CER) caused by noise is the only factor in the experiment. It also compares the different effects of the burst error model and the independent error model on TCP performance.

- The effect of TCP packet size: TCP performance degrades over ATM networks because of the segmentation and reassembly process due to the size mismatch of TCP packets and ATM cells. It implies that larger TCP packet sizes result in poorer TCP performance. This experiment is designed to study how the change of TCP segment sizes affects the performance.

- The effect of bandwidth asymmetry: Service providers offer different ADSL line speeds. The downstream and upstream transmission rates are asymmetric. Under the same error ratio, lines of different transmission rates are affected to a different extent by errors. This experiment varies the downstream and upstream transmission rates of the ADSL lines and studies the impact on TCP performance.

- The effect of percentage of noisy lines: On a real network, it is likely that during a certain period some of the lines are noisy, while others are in good status. Service providers need to know how many noisy lines they can afford on their networks to achieve competitive performance. This experiment is designed to study how the change in percentage of noisy lines affects network performance. In the experiment, the number of noisy lines is the major factor. The number of noisy lines varies from zero to eight in each experiment.

- The effect of switch buffer size: This experiment is used to explore TCP performance in the presence of both noise and congestion. The preceding experiments are based on a network without data loss by congestion in order to isolate and focus on the effect of noise. However, congestion is inevitable in the real world and it is one of the major factors that causes data loss. In this experiment, switch buffer size and noise on the lines are the two variables being varied. When the switch buffer size is small, the network experiences loss due both to noise and to congestion.

4.6 Summary

This chapter has presented the experimental design for the performance study of TCP/ATM/ADSL. First, it introduced the TCP traffic model that was used in the experiments. Second, it illustrated the network structure and listed the setup of parameters. Third, it defined and explained the performance metrics that were used to evaluate the experimental results. Finally, it described the design of each group of experiments in terms of their motivations and objectives.

Chapter 5

Experimental Results and Evaluation

This chapter presents and analyzes the experimental results of TCP over ATM over lossy asymmetric networks. The chapter starts with an illustration of the dynamic behavior of TCP under loss in Section 5.1. Section 5.2 studies the effect of noise on TCP performance in isolation. Section 5.3 compares TCP's reaction to loss using different TCP segment sizes. Section 5.4 varies the downstream and upstream transmission rates and explores their effects on TCP performance. Section 5.5 studies the network performance when the percentage of noisy lines within a network changes. Section 5.6 investigates the effects of both noise and congestion on TCP performance. Finally, Section 5.7 summarizes the findings from the experimental results.

5.1 Dynamics of TCP

Experiments have been conducted with the single TCP source scenario to reveal the dynamic behavior of TCP in the presence of transmission errors. In order to study the effect of noise in isolation, large switch buffers were used to avoid buffer overflows due to congestion. However, all the TCP mechanisms for congestion control and flow control apply because TCP reacts to errors in the same way as it reacts to congestion.

5.1.1 TCP Congestion Control Algorithms

There are four intertwined algorithms for TCP congestion control: slow start, congestion avoidance, fast retransmit, and fast recovery. Here is a brief description of how these algorithms work [2] [30] [31] [34].

Slow start is used by the TCP sender to control the amount of data being injected into the network. Slow start adds another window to TCP: the congestion window, called "cwnd". The congestion window is the flow control variable at the sender side, while the receiver's advertised window (rwnd) is the flow control variable imposed by the receiver. The TCP sender can never send an amount of data that exceeds the minimum of the sender's congestion window and the receiver's advertised window. Each time a new TCP connection is established, the congestion window is initialized to one TCP segment. For each acknowledgment received, the congestion window is increased by one segment. The sender starts by transmitting one segment. When this segment is acknowledged, the congestion window is increased from one to two, and two segments are sent. When each of these two segments is acknowledged, the congestion window is increased from two segments to four. This enables the size of the congestion window to grow exponentially. Additionally, when a retransmission timeout occurs during a connection due to one or more lost packets, the congestion window size is reset to one TCP segment and TCP starts from slow start again.

Congestion avoidance is another algorithm for congestion control which is usually implemented together with slow start. Congestion avoidance maintains a threshold, which is initially set to 65,535 bytes. There are two possible indications of packet loss: the occurrence of a timeout, and the receipt of duplicate ACKs. When packet loss is detected, one-half of the current window size (the minimum of cwnd and rwnd, but at least two segments) is saved as the threshold. If cwnd is less than or equal to the threshold, TCP performs slow start; otherwise, congestion avoidance is used. Slow start increases the congestion window size exponentially. When the congestion window is larger than the threshold, congestion avoidance takes over. It increases the congestion window linearly by an increment of MSS*MSS/cwnd each time an ACK is received [19] [31]. The combination of slow start and congestion avoidance enables TCP to grow fast from a small window size and become more conservative when the window size is large.

The fast retransmit algorithm allows TCP to react to data loss quickly, thus avoiding coarse timeouts. When an out-of-order segment arrives, the TCP receiver should send an immediate duplicate ACK for a previously received segment. This ACK informs the sender that a segment was received out of order and which sequence number is expected. Duplicate ACKs can be caused by packet loss or reordering. When three or more duplicate ACKs are received consecutively for the same sequence number, it is a strong indication that a segment has been lost. Upon receiving duplicate ACKs, the sender will immediately retransmit the missing segment without waiting for a retransmission timer to expire. This algorithm is called "fast retransmit".

After the sender sends the missing segment using the fast retransmit algorithm, congestion avoidance (not slow start) is performed. This is the TCP "fast recovery" algorithm. It allows high throughput under moderate packet loss.

5.1.2 TCP Congestion Window Dynamics

The following graphs depict the change in the TCP congestion window as a function of time for lossy networks with different values for the Cell Error Ratio (CER).

Figure 5.1 shows the congestion window for the burst error model when the CER varies from 1.0×10^{-3} to 1.0×10^{-5}. The top-left graph shows that the TCP congestion window can hardly grow because of frequent packet loss when CER = 1.0×10^{-3}. The graph below it illustrates packet drops in the same time scale for the same CER value. At about time 4, two consecutive packets have been lost. This causes a timeout, shown as the idle period in the congestion window plot. A following packet drop results in an even longer timeout from about time 6 to 10. Another consecutive packet loss at a later time has been recovered by fast retransmit. It is seen that it is hard for TCP to recover from consecutive packet loss by fast retransmit, especially when the window size is small.

The plot for CER = 1.0×10^{-4} illustrates the slow start and fast retransmit processes clearly. Each time when packet loss occurs and the retransmission timer expires, the congestion window is reset to one packet, and TCP is forced to begin

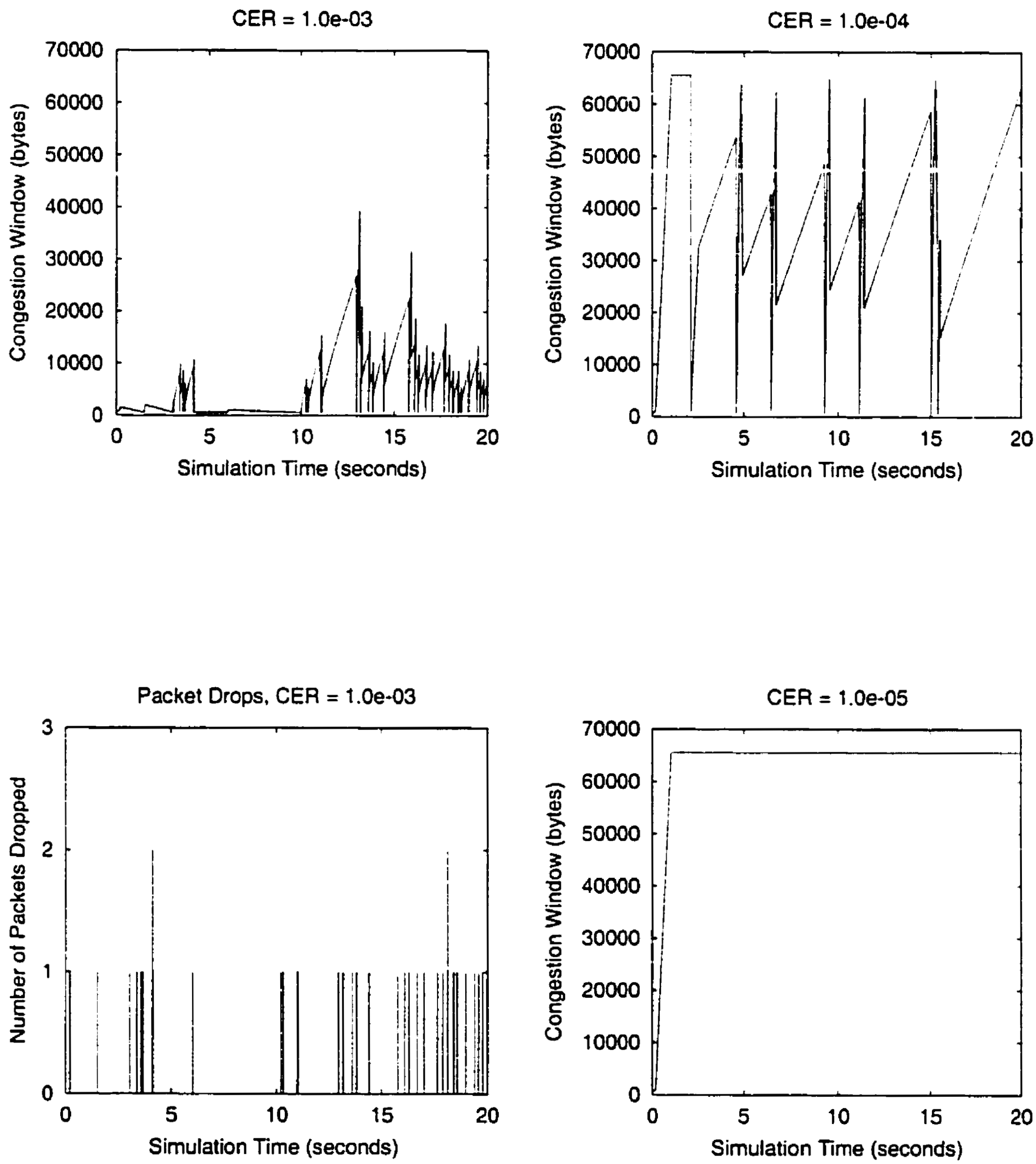


Figure 5.1: Congestion Window Dynamics of the Single TCP Source Scenario (Burst Error Model)

with slow start. The fast growing of the congestion window size shows that the slow start algorithm is not "slow" because it increases the congestion window size exponentially. Even so, it reduces the throughput of TCP greatly by dropping the congestion window to one packet. When the congestion window reaches half of its value before slow start, the speed of growth is slowed down because the congestion avoidance algorithm is in control. The fast retransmit and fast recovery process is illustrated when the congestion window shrinks to one-half and grows linearly again. When CER = 1.0×10^{-5}, no packet loss occurs during the short simulation time (20 seconds). The congestion window grows quickly and maintains the maximum permissible window size (65,535 bytes) imposed by the receiver. This illustrates the ideal condition for TCP when there is no packet loss.

Figure 5.2 shows the congestion window changes for the independent error model, for CER values ranging from 1.0×10^{-3} to 1.0×10^{-6}. It yields similar results to the burst error model. When CER = 1.0×10^{-3}, both error models cause problems for TCP. The TCP congestion window cannot be opened because there are too many successive retransmissions and timeouts.

Figure 5.1 and 5.2 show that fast retransmit and fast recovery can help TCP react to packet loss in a timely manner compared with coarse timeouts. However, these algorithms cannot eliminate all coarse-grained timeouts because of the following reasons. First, ACKs may also be lost, especially on a very lossy network. If the sender does not receive acknowledgments in time, the fast retransmit cannot be triggered and the retransmission timer will be in control. When the retransmission timer expires, TCP will resume from slow start. This is the case in this study because the reverse channel for ACKs has the same CER as the data packet channel.

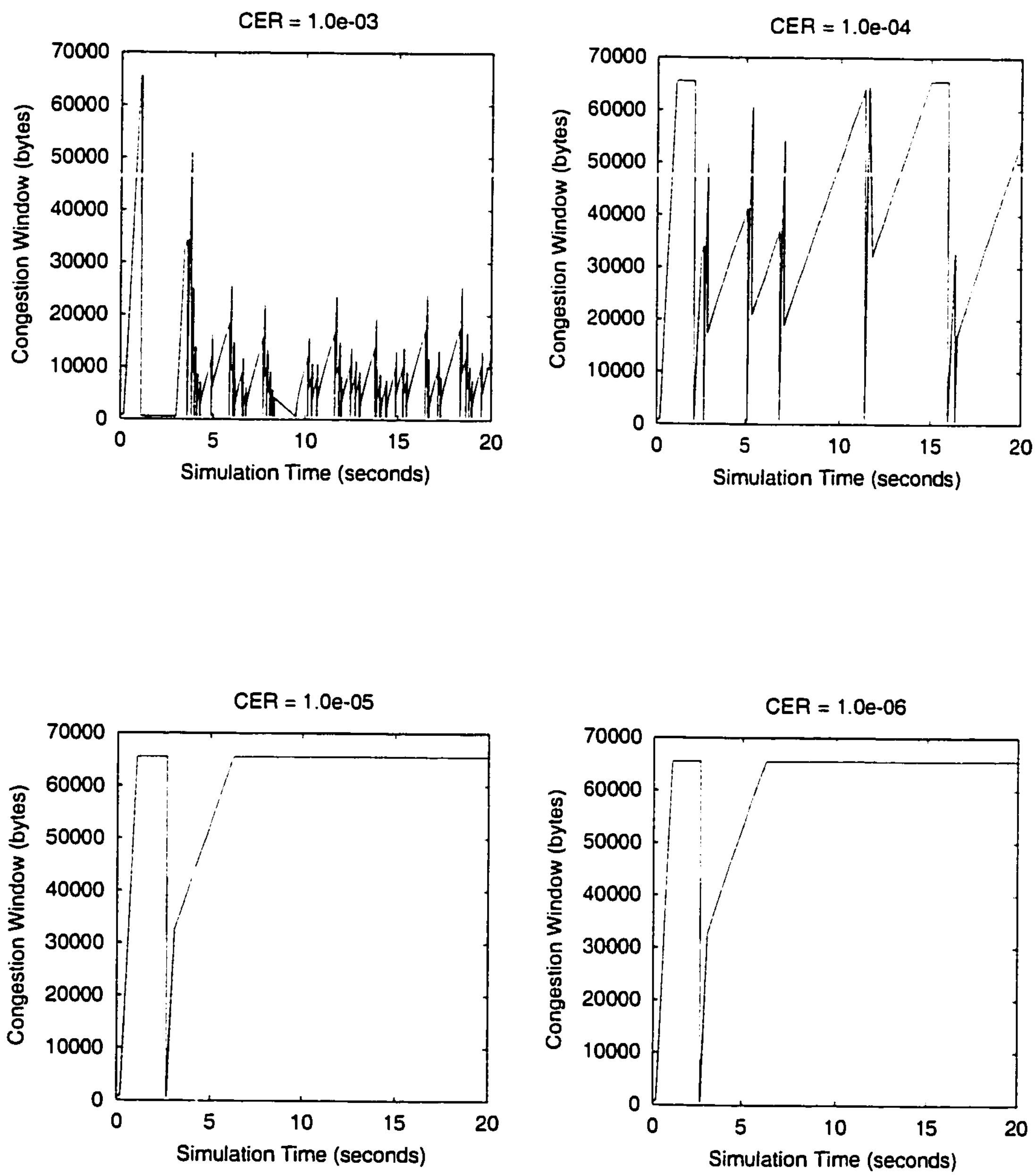


Figure 5.2: Congestion Window Dynamics of the Single TCP Source Scenario (Independent Error Model)

The second reason is that TCP's transmission window is usually not large enough to contain duplicate ACKs tp trigger fast retransmits when multiple packets have been lost [7]. This is why TCP performance can be severely degraded under the loss of multiple consecutive packets.

Figure 5.1 and 5.2 also illustrate the additive increase and multiplicative decrease feature of TCP congestion control. A TCP source sets the congestion window based on the level of congestion it perceives on the network. In this thesis work, the congestion window is affected by errors on the ADSL lines. When there is no packet loss, the congestion window is increased exponentially in the slow start phase or linearly in the congestion avoidance phase; when packet loss occurs, the congestion window is decreased by half (fast recovery) or even dropped to one packet (slow start). This is called the additive increase and multiplicative decrease of TCP. It enables TCP to transmit packets at the allowable rate of the network and shrink quickly to avoid more data loss when congestion (or error) happens.

Table 5.1 shows the achieved throughput in bits per second by the single TCP source for both error models. The simulation is run for 50,000 seconds, long enough for the network to reach steady state. According to Equation 4.1, the maximum achievable throughput for a link with a bandwidth of 1,500,000 bits per second should be 1,207,547 bits per second. When CER is low, the rare errors on the network only have minimal impact on the throughput because most of the packet loss can be recovered using fast retransmits. When CER is high, the number of coarse timeouts increases and the throughput obviously decreases.

Table 5.1: Throughput (bits/sec) under Various CERs

CER	Burst Error Model	Independent Error Model
1.0×10^{-3}	1,012,644	951,304
1.0×10^{-4}	1,187,034	1,188,322
1.0×10^{-5}	1,205,912	1,206,739
1.0×10^{-6}	1,207,101	1,207,480
1.0×10^{-7}	1,207,544	1,207,545

5.1.3 TCP Sequence Number Analysis

TCP guarantees the delivery of packets by assigning each a sequence number. The acknowledgment of sequence number n tells the sender that the next expected sequence number is n (i.e., all data prior to n has been received successfully). When packet loss occurs, the receiver sends duplicate acknowledgments, and the sender resends the missing packet with the required sequence number. The behavior of packet loss and recovery can be illustrated clearly by plotting the TCP sequence numbers over simulation time.

Figure 5.3 and 5.4 depict the TCP sequence numbers for the burst and independent error models when CER varies from 1.0×10^{-3} to 1.0×10^{-5}. The TCP sequence numbers and ACK numbers are drawn in separate graphs for clarity. In the sequence number plot, retransmission is demonstrated by the transmission of a packet with the same sequence number as a preceding packet, received at a later time. In the ACK number plot, packet loss is expressed by a horizontal portion on the curve, which indicates duplicate ACKs. When CER $= 1.0 \times 10^{-3}$, the scales for the horizontal and vertical axes are different from those in the plots for other

CER values in order to illustrate the packet loss and retransmissions clearly.

The packet loss and recovery are illustrated clearly in the plots for CER = 1.0×10^{-4}. Each time when a packet is lost, the TCP receiver starts sending duplicate ACKs, shown as the small horizontal portion on the curve. When the TCP sender gets these duplicate ACKs for the same sequence number, it resends the packet with this sequence number. Upon receiving the resent packet, the TCP receiver sends the ACK with the next expected sequence number. Several packet loss and recovery episodes have been shown in the plots. When CER = 1.0×10^{-5}, the sequence number increases without any breaks for the burst error model because there is no packet loss in the short simulation time (10 seconds).

Since the horizontal axis represents time and the vertical axis represents the cumulative data being transmitted, the slope of the TCP sequence number curve is the throughput. Comparing the plots for CER = 1.0×10^{-4} and CER = 1.0×10^{-5}, it can be observed that the throughput increases when the error ratio decreases as expected.

The TCP congestion window and sequence number are two variables that represent two views of TCP dynamic behavior. For example, packet loss is indicated as a break in the TCP sequence number plot, a small horizontal portion in the TCP ACK number plot, and a sharp decrease of the congestion window. This can be seen by comparing the TCP congestion window plots and the sequence/ACK number plots for the same error model and the same CER.

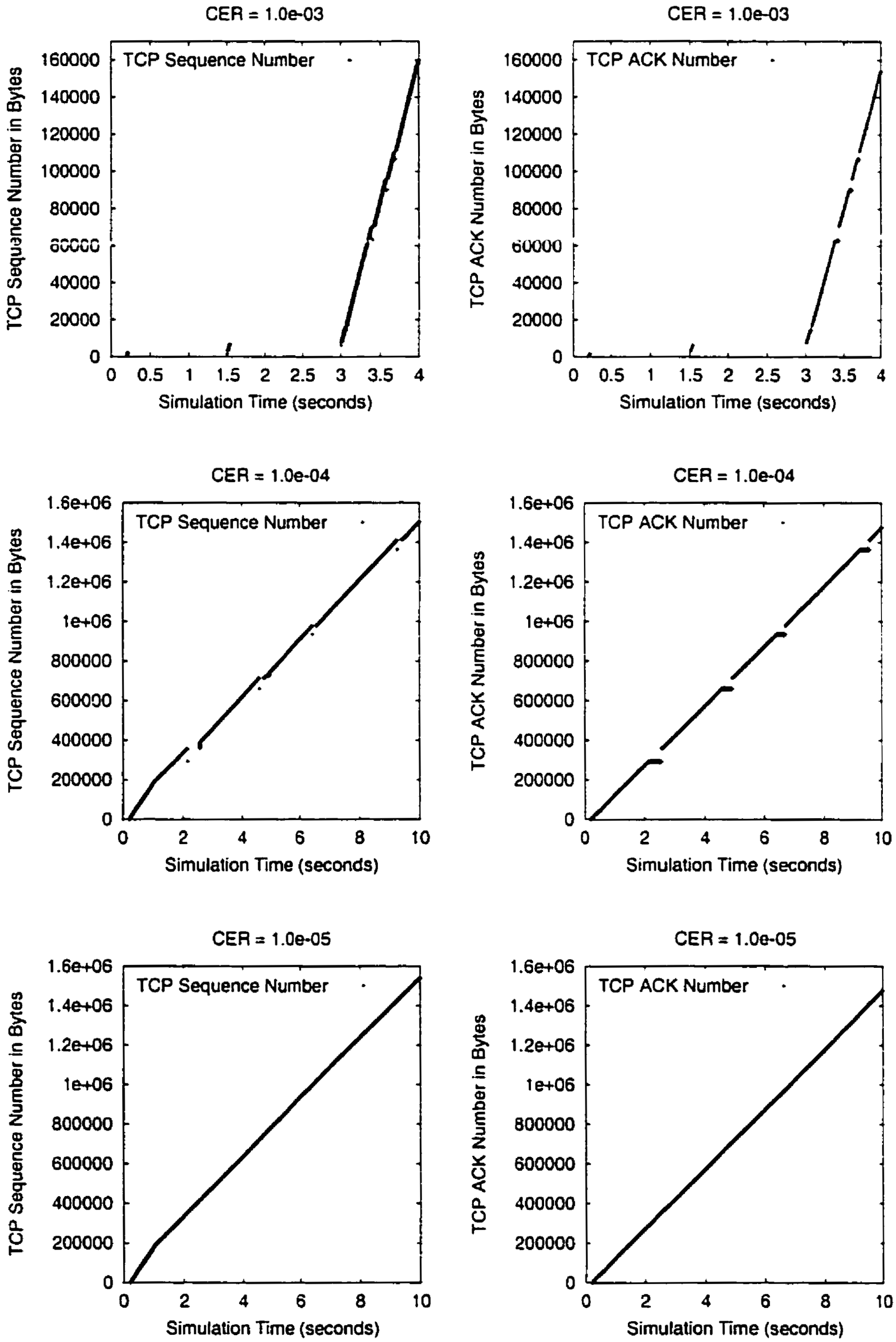


Figure 5.3: TCP SEQ/ACK Numbers of the Single TCP Source Scenario (Burst Error Model)

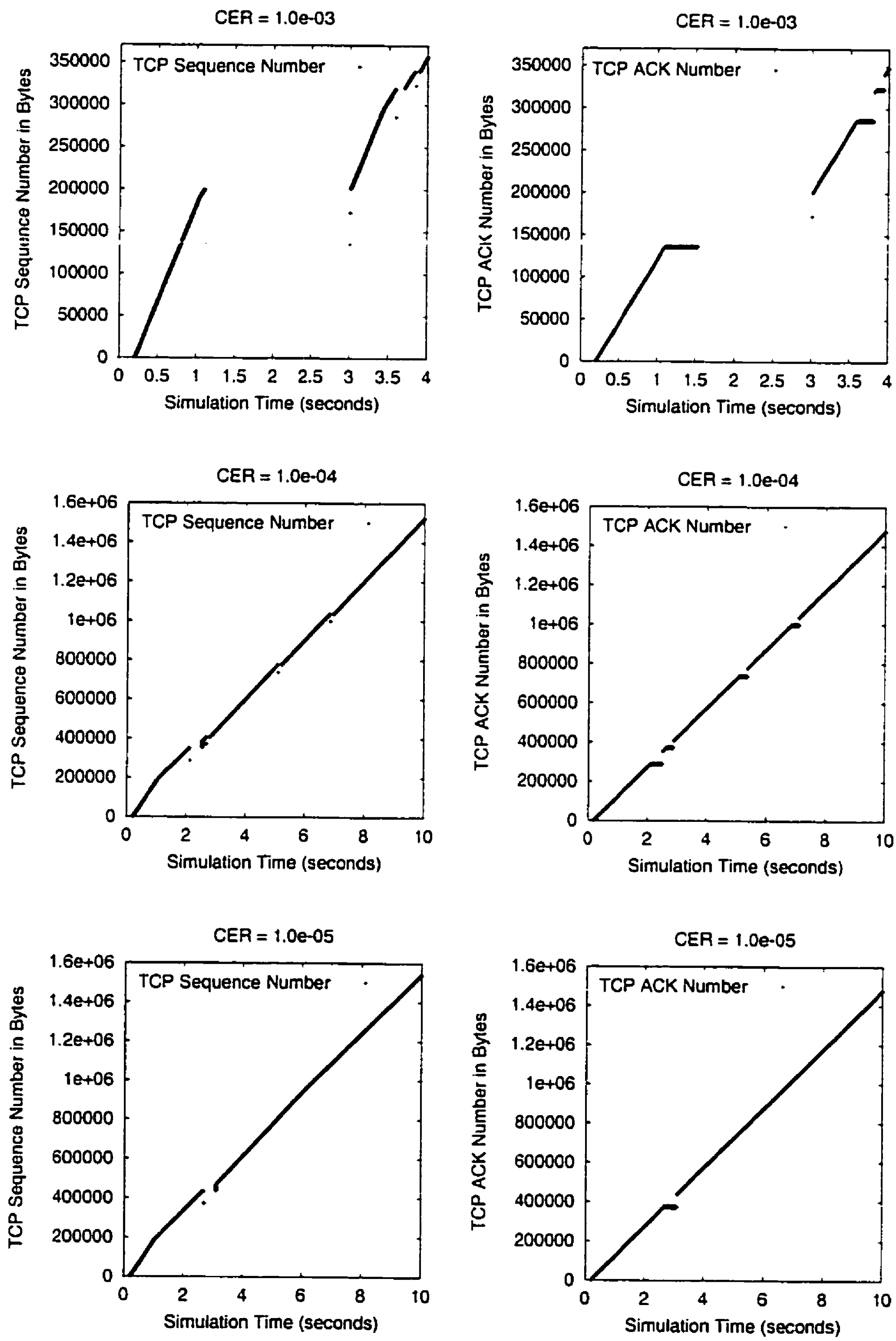


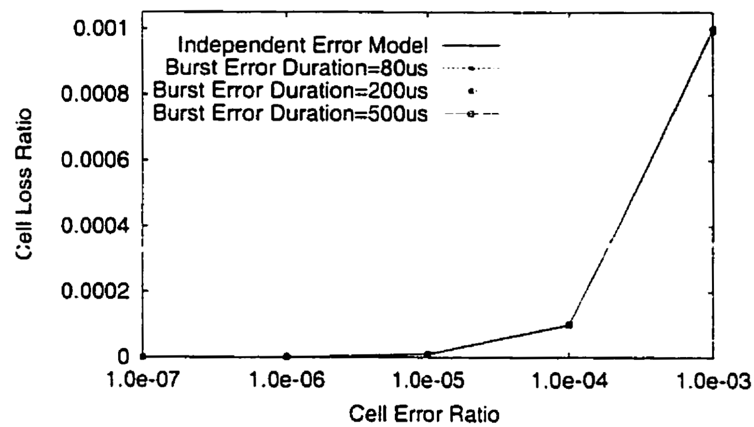
Figure 5.4: TCP SEQ/ACK Numbers of the Single TCP Source Scenario (Independent Error Model)

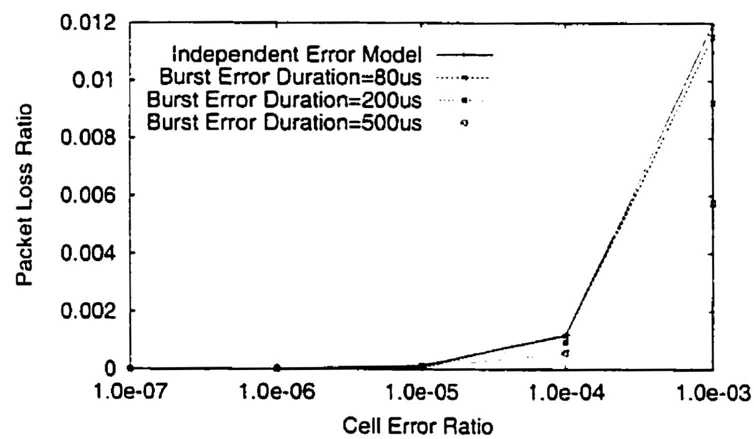
5.2 Effect of Noise

The effect of data loss due to noise is the major focus of the thesis. This experiment is designed to study how the Cell Error Ratio (CER) affects the end-to-end TCP performance. The range of CER is from 1.0×10^{-3} to 1.0×10^{-7}. For the burst error model, three mean burst error durations were used: 80 μs, 200 μs, and 500 μs. The purpose is to explore how the two error models affect TCP performance. Two levels of data loss are studied: the cell loss at the ATM layer and the packet loss at the TCP layer. Effective throughput is another performance metric that is studied. The results are shown in Figure 5.5.

As designed, the Cell Loss Ratio (CLR) is always close to the CER. CER is the targeted cell loss ratio which is defined by the simulation input parameter. CLR is the observed cell loss ratio from the simulation statistics. The observed CLR should be the same as the targeted CER. An experiment with the single TCP source scenario comparing the targeted and observed data has been done in Subsection 3.3.2. The results of this experiment can also be used as validation of the error models. The different error models and burst error durations do not affect the cell loss ratio. They only decide the distribution of errors. This is why CLR is the same for all the scenarios. For clarity, a logarithmic scale is used for the horizontal axis. The vertical axis does not use the logarithmic scale so that the trend of the curve can be observed easily. Plots with logarithmic scales for both the horizontal and vertical axes can be found in Figure 3.7.

The different error distribution patterns do make a difference on the Packet Loss Ratio (PLR). The burst error model always has a lower PLR than the independent error model. When the mean burst error duration increases, the difference in PLR





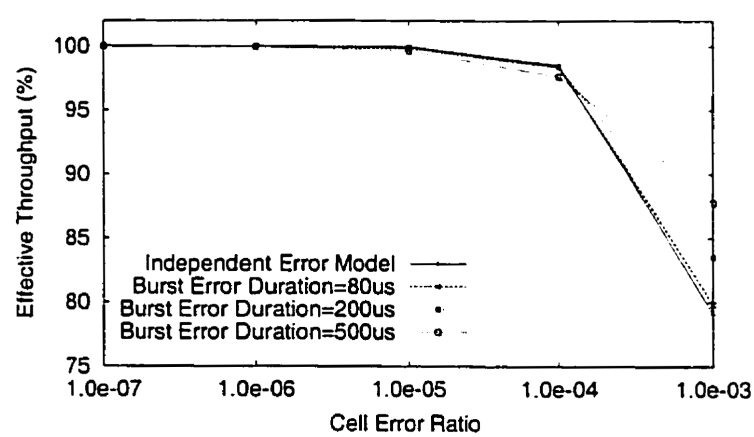


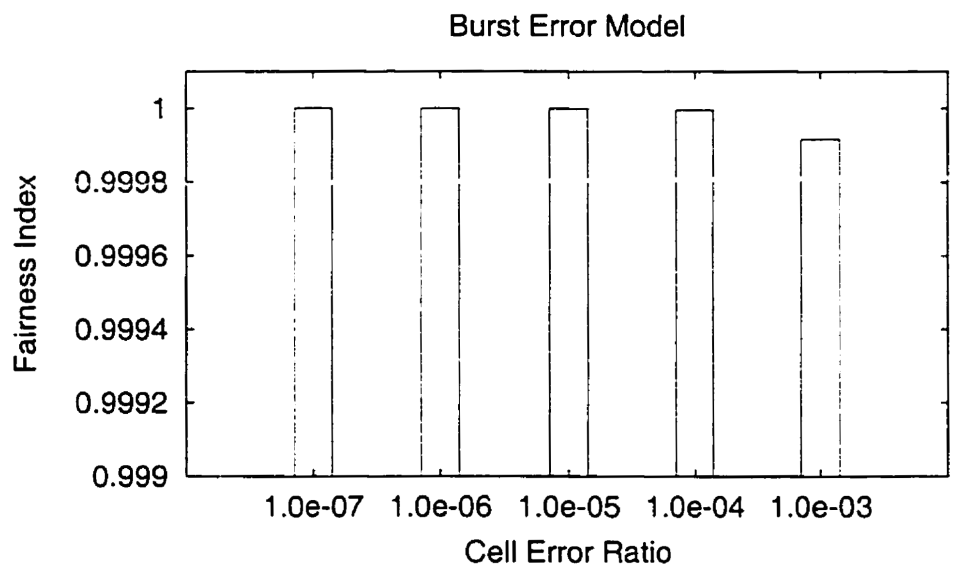
Figure 5.5: Effect of Cell Error Ratio on CLR, PLR, and Effective Throughput

becomes more significant. The reason is intuitive. The errors in the independent model are scattered and each corrupt cell destroys one packet. The errors in the burst error model occur in clumps. Several corrupt cells may only destroy one packet. So with the same number of corrupt cells, the independent error model destroys more packets than the burst error model. When the mean burst error duration increases, the intervals between bursts increase and the average number of corrupt cells in a corrupt packet becomes larger. This is why the burst error model with the longest mean burst error duration has the lowest PLR in the plot.

When CER is low, the effective throughput for all the scenarios approaches 100%. There are two factors that affect the throughput. One is the frequency of occurrence of the errors. The other is the duration of a burst of errors. If errors appear frequently, they cause severe "oscillation", i.e., the congestion window suffers frequent sudden drops because of cell loss. If the burst duration of errors is long, it causes multiple packet loss. This could result in timeouts and reduce TCP throughput. When CER is lower than 1.0×10^{-4}, the effective throughput of the independent error model is slightly higher than the burst error model. This indicates that when error occurrence is not frequent, the scattered errors have less effect on TCP throughput than the bursts of errors because TCP can recover faster from single packet loss than from multiple packet loss by using fast retransmits. However, when CER is higher than 1.0×10^{-4}, the TCP throughput is higher for the burst error model. This indicates that the severe "oscillation" is harmful because the window cannot be maintained at a steady size. The multiple packet loss is relatively "better" because the window can grow during the intervals between error bursts.

Figure 5.6 shows the fairness index for the eight TCP sources. It can be seen that the scenario achieved good fairness in all the experiments. All of the fairness indices are very close to 1. The fairness index in all the experiments of this thesis study is high and will not be plotted for the following experiments unless needed. One reason for good fairness is that congestion occurs separately on each ADSL line, not on the shared link between the ATM switch and ADSL Access Node. Another reason is that all TCP sources are configured the same. There is no greedy source to acquire network bandwidth.

Figure 5.7 compares the PLR and effective throughput of LAN and WAN scenarios in the presence of errors. As shown in Figure 4.3, the distance between the ADSL Access Node and ATM switch in the core ATM network is 5 Km for LAN and 1,000 Km for WAN. Other parameters are the same for LAN and WAN. WAN has a longer average round-trip time (RTT) than LAN. The longer RTT has no effect on PLR since PLR is only affected by the error ratio and error distribution. It has a slight effect on the effective throughput. When CER is low, the effective throughput is 100% in WAN, the same as that in LAN. When CER is high, the effective throughput for WAN is a bit lower than that for LAN. This is because the calculation of TCP's retransmission timeout is related to RTT. A longer RTT makes TCP react slower to packet loss. However, the throughput difference of LAN and WAN is negligible.



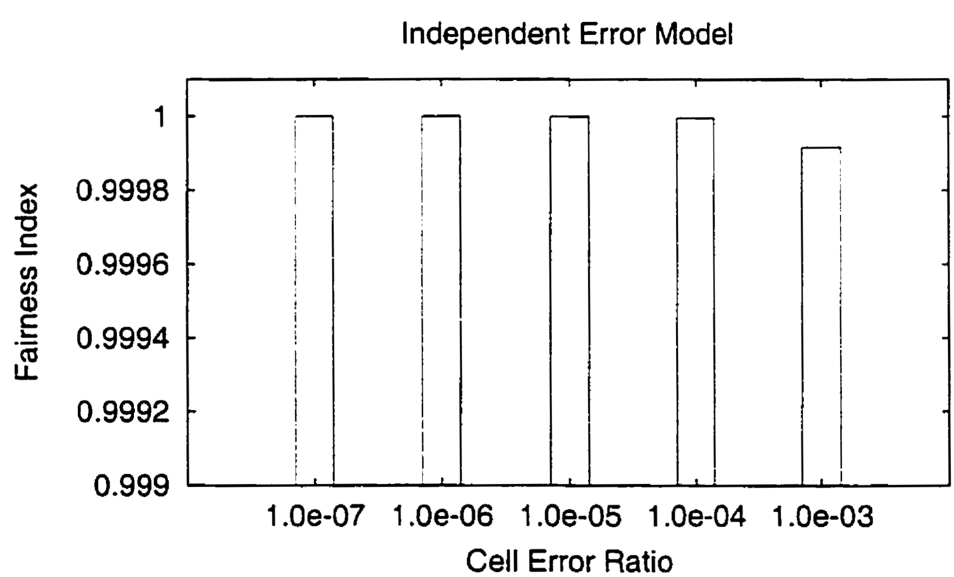


Figure 5.6: Fairness Index for the Eight TCP Sources Scenario

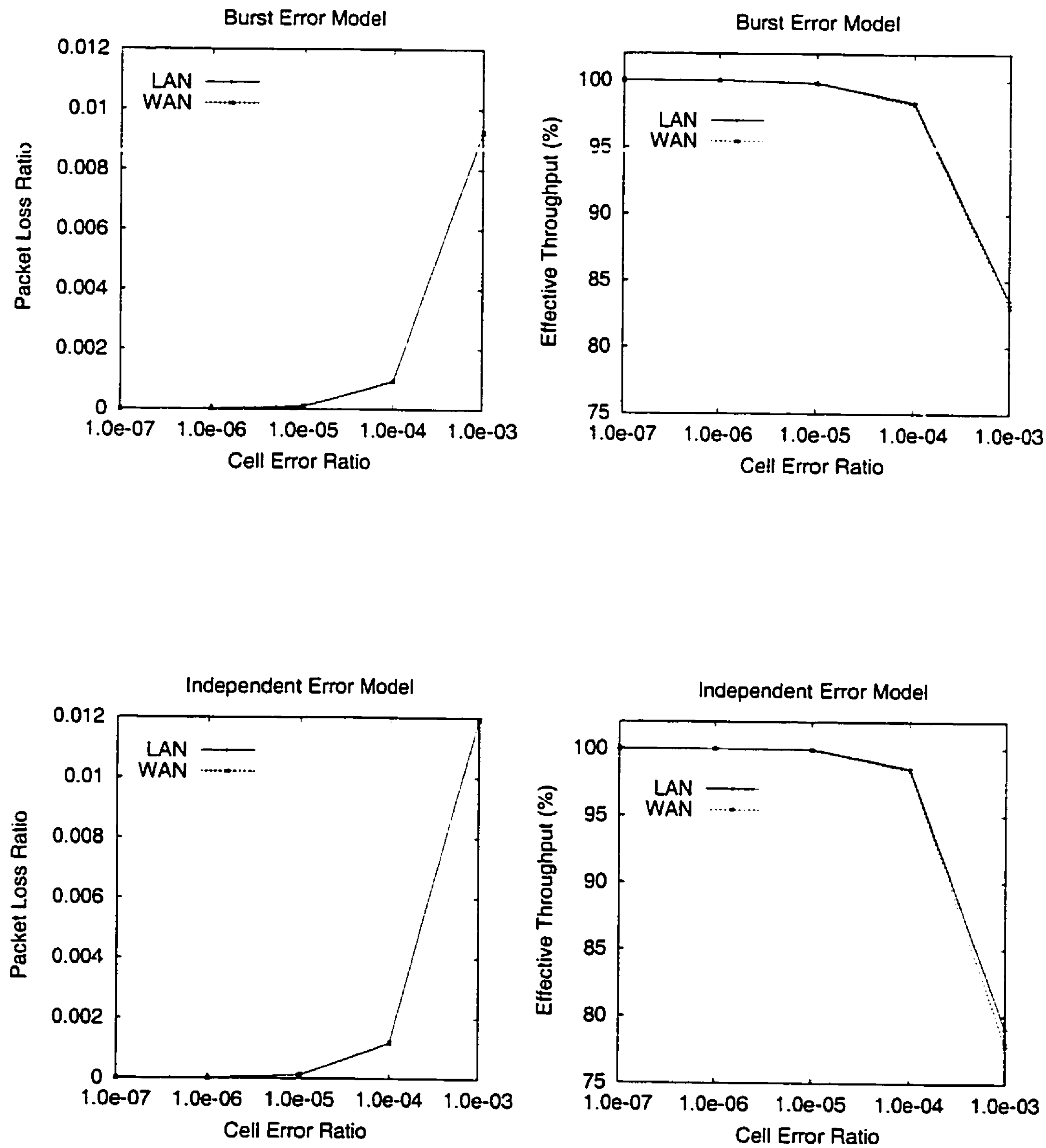


Figure 5.7: Performance Comparison of LAN and WAN

5.3 Effect of TCP Packet Size

TCP packet size (MSS) is another important factor that affects TCP performance. Figure 5.8 and 5.9 show the CLR, PLR, and effective throughput as a function of CER and MSS for the burst error model and the independent error model respectively. Three packet sizes are studied: 512 bytes used for the Internet, 4352 bytes for FDDI, and 9140 bytes for IP over ATM. According to Equation 4.1, a 512-byte packet requires 12 ATM cells, a 4352-byte packet requires 92 cells, and a 9140-byte packet requires 192 cells.

As expected, the CLR plots show that the different MSS values make no difference to CLR. The reason is that CLR should always be identical to CER theoretically. CLR is the measurement of cell loss at the ATM layer. The change of packet size at the TCP/IP layer will not affect the cell loss ratio at the ATM layer.

The impact of MSS is shown clearly in the PLR plots. For both error models, the larger the MSS, the higher the packet loss ratio is. The effect is more obvious when CER is higher. According to [7], a larger packet increases packet loss rate. For example, if the loss probability of 1 KB packets is 10%, the loss probability of 0.5 KB packets will be between 5% and 10% [7]. Another reason is that under the same transmission rate, fewer packets are sent with large MSS in a certain period. So the denominator while calculating PLR becomes smaller. The increase of PLR along with the increase of MSS can adversely affect network performance. In the PLR plot for the independent error model, when CER $= 1.0 \times 10^{-3}$, the PLR for MSS = 512 bytes is about 1% , while the PLR for MSS = 9140 is about 17%, more than ten times higher. A network with such a high PLR could be very unreliable.

Studies show that large packet sizes can reduce the overall effective throughput

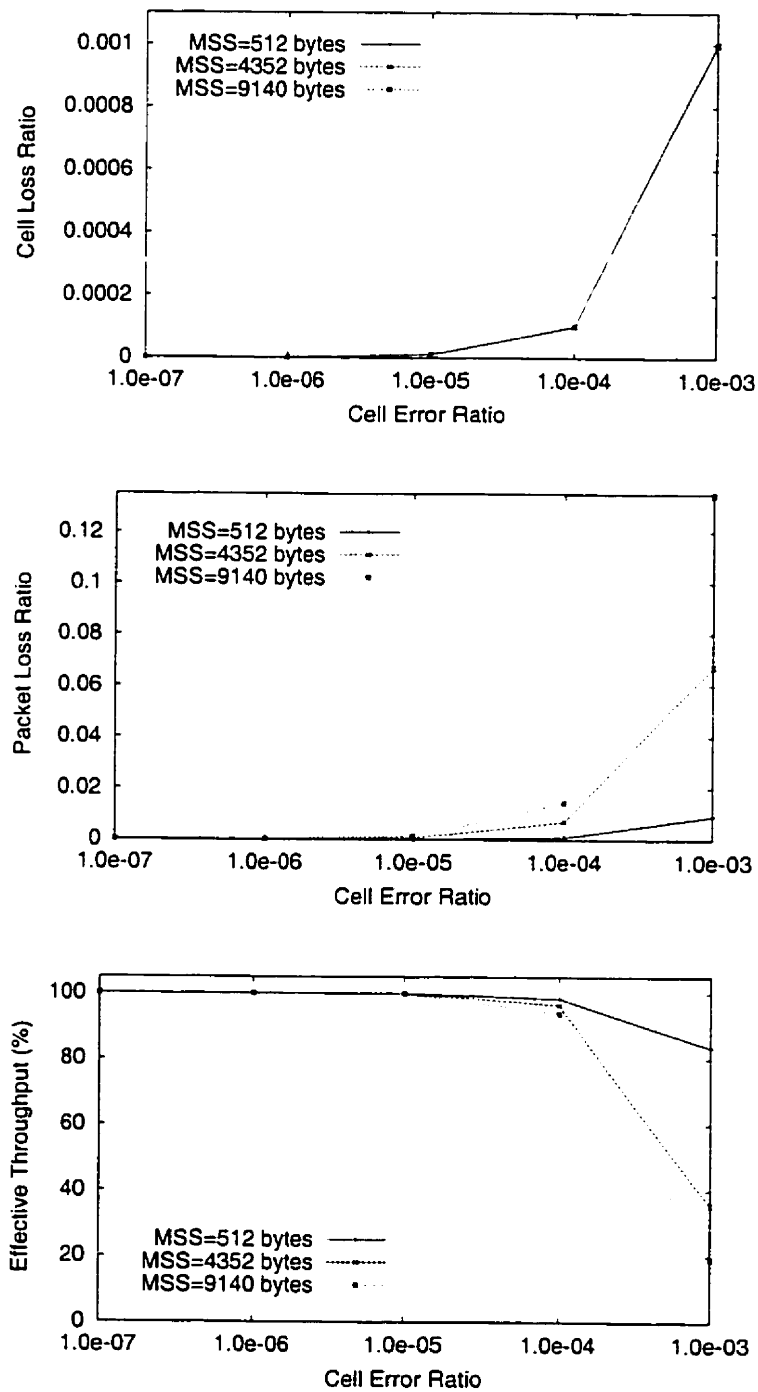
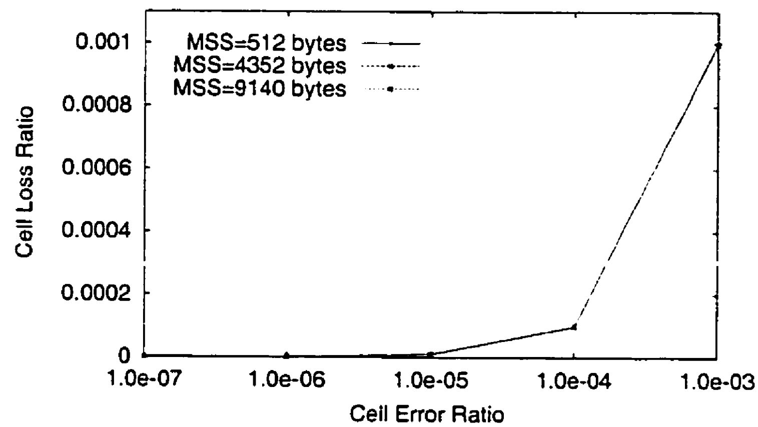
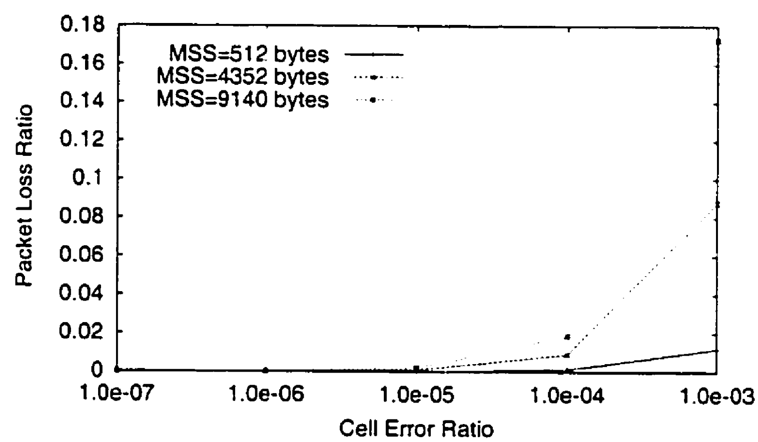


Figure 5.8: Effect of Packet Size on CLR, PLR, and Effective Throughput (Burst Error Model)





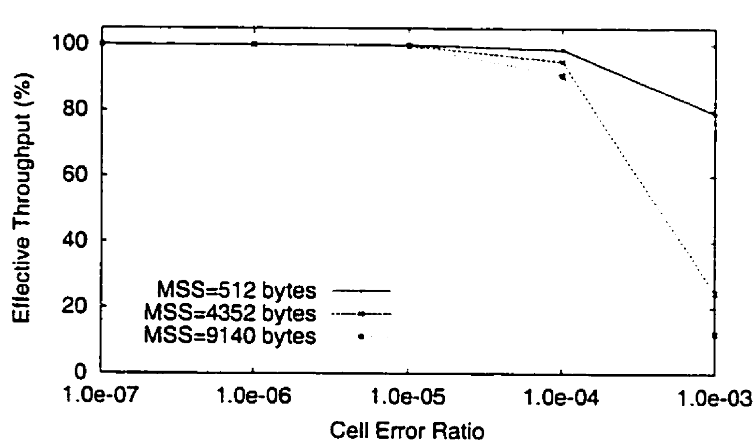


Figure 5.9: Effect of Packet Size on CLR, PLR, and Effective Throughput (Independent Error Model)

[7] [28]. One reason is that large packet sizes increase the number of wasted cells transmitted on the network when a cell from one packet is dropped. Another reason is that fewer ACKs can be held in the TCP window with larger packet sizes. If duplicate ACKs cannot be received correctly, fast retransmit cannot be triggered and TCP will start transmitting with a congestion window size of one packet. Therefore, TCP cannot detect and react to loss quickly. In the plots, the difference of effective throughput among the three MSS values becomes more obvious when CER = 1.0×10^{-3}. It indicates that large packets make TCP more vulnerable to errors.

Several tradeoffs need to be considered when selecting the MSS values. With a smaller MSS value, TCP is more resilient in reacting to packet loss. The TCP fast retransmit and recovery mechanisms can be easily triggered because TCP can receive more ACKs with a smaller packet size. The negative side of smaller packet sizes is the greater TCP/IP header overhead. TCP's slow start and congestion avoidance mechanisms are also slowed down. However, TCP is usually more efficient and robust using smaller packets than using larger packets.

5.4 Effect of Bandwidth Asymmetry

Bandwidth asymmetry is an important characteristic and considered to be an advantage of ADSL. Many network operations, such as Internet access, are inherently asymmetric. The effect of bandwidth asymmetry on network performance is studied in this section. In the experiments, the upstream (ACKs) rate is fixed at 512 Kbps and the downstream rate (data) is varied from 1.5 Mbps to 8.0 Mbps. Figure 5.10

and 5.11 show the results for the burst and independent error models.

For both error models, CLR is not affected by transmission rates. It is because CLR is only decided by CER. It is not a function of bandwidth. For a certain CER value, when transmission rate increases, more cells are transmitted within a period of time, and more cells are corrupted. Thus the ratio of cell loss does not change.

For the burst error model, PLR decreases when transmission rate increases. However, PLR is not affected by transmission rates in the independent model. This can be explained by studying the "nature" of the two error models. The burst error model is defined in the time domain, and the occurrence of bursts is independent of the data transmission, i.e., there could be bursts of errors whether the line is busy or idle. Figure 5.12 illustrates what happened by drawing an example of packet drops for two different transmission rates. The rectangles filled with light color represent packets, and the ones shaded with stripes represent noise. When transmission rate increases, the time to transmit one packet becomes shorter and more packets are transmitted within a time unit. The occurrence of noise is the same for the two rates. In the example shown in Figure 5.12, $\frac{1}{2}$ of the total packets are corrupted by the two bursts of noise when the transmission rate is low, while $\frac{7}{15}$ packets are corrupted when the transmission rate is high. More experiments with transmission rates higher than 8 Mbps also show that a higher transmission rate results in a lower PLR for the burst error model. In the independent error model, cell loss is based on each incoming cell. When the total number of packets increases, the total number of corrupt packets also increases according to the CER. This is why in the independent model PLR remains unchanged for different transmission rates.

The effective throughput plots show that when transmission rate increases, effec-

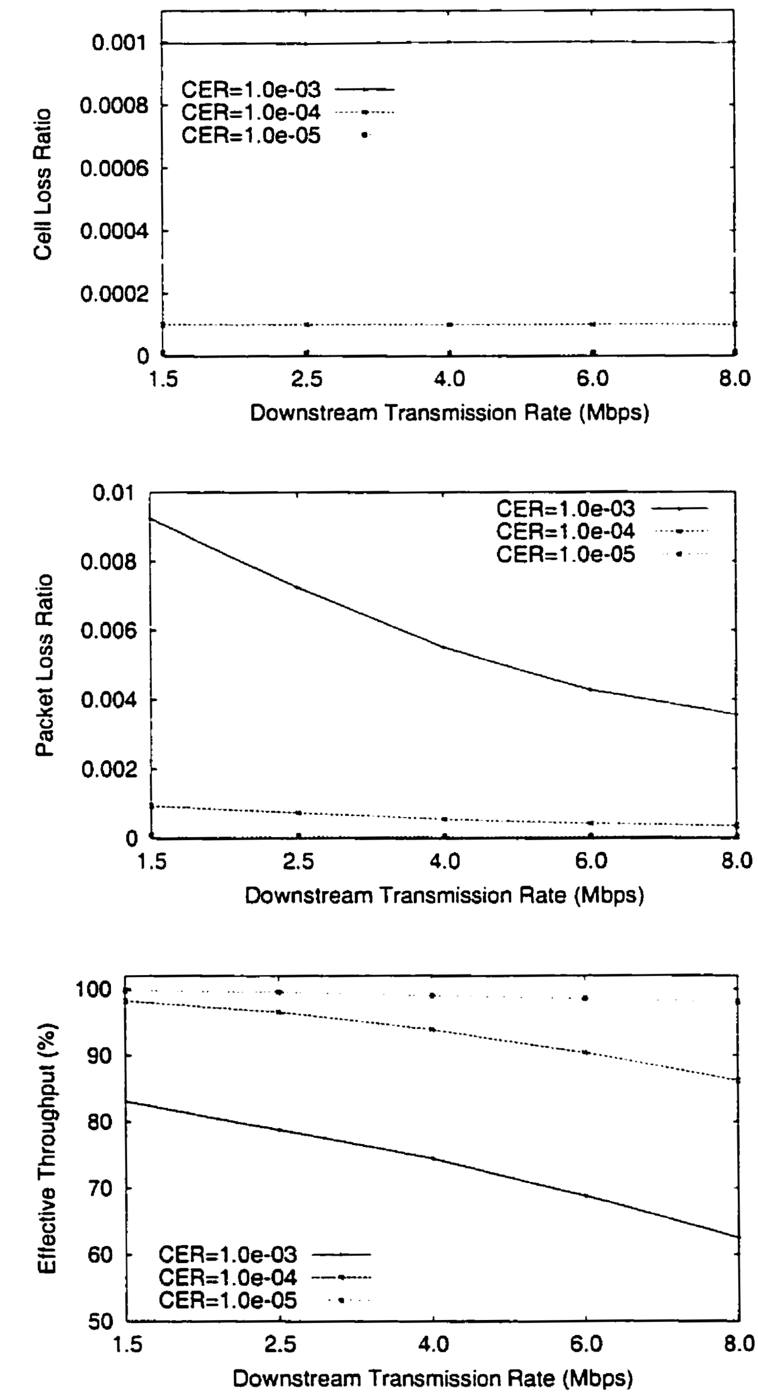


Figure 5.10: Effect of Bandwidth Asymmetry on CLR, PLR, and Effective Throughput (Burst Error Model), downstream rate: varies; upstream rate: 0.512 Mbps

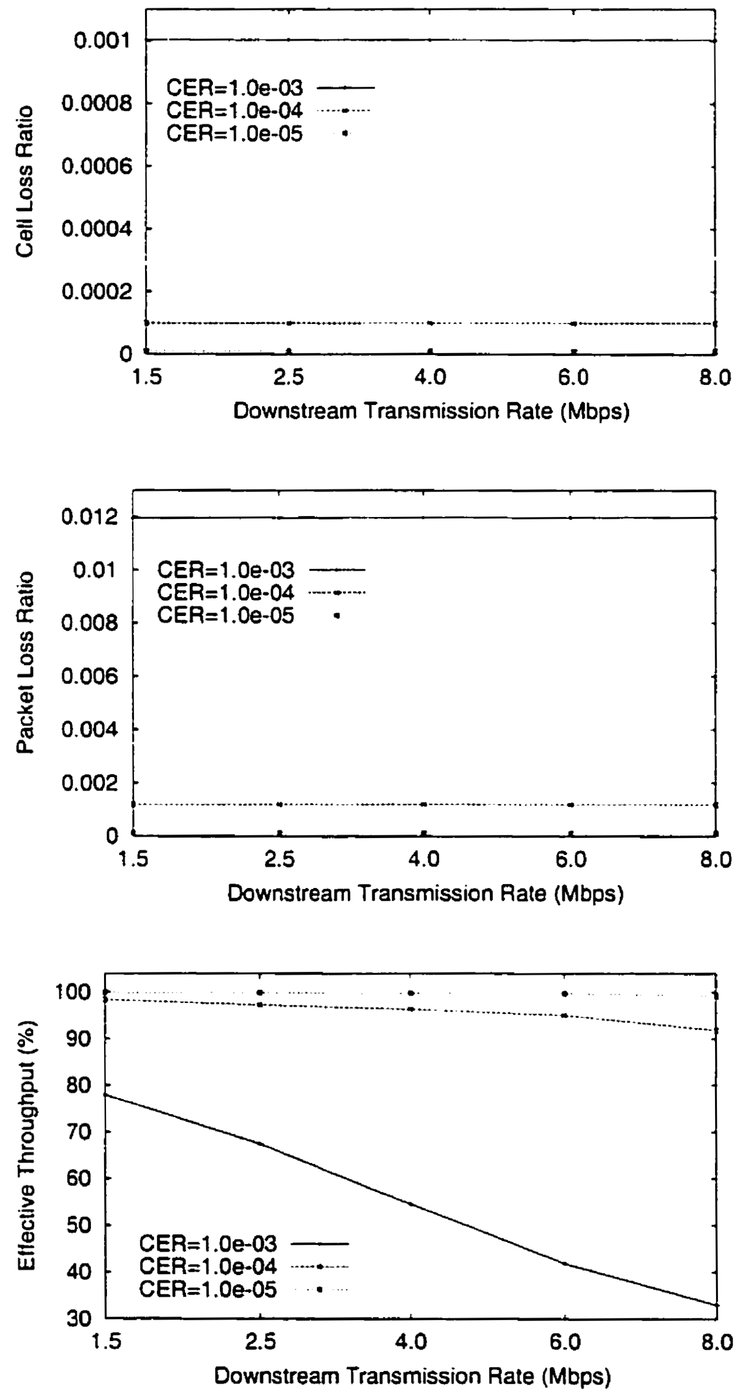


Figure 5.11: Effect of Bandwidth Asymmetry on CLR, PLR, and Effective Throughput (Independent Error Model), downstream rate: varies; upstream rate: 0.512 Mbps

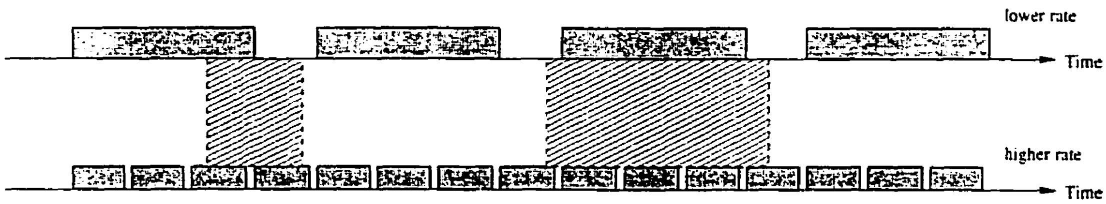


Figure 5.12: Packet Loss Illustration for Different Transmission Rates (burst error model)

tive throughput decreases. The phenomenon is more obvious when CER is higher. One reason is that higher transmission rates make the "oscillation" of the TCP window more severe when data loss occurs. Figure 5.13 compares the change in the congestion window size with transmission rates of 1.5 Mbps and 8.0 Mbps for the independent error model. Although the window can grow faster when the rate is higher, it shrinks more frequently because of cell loss. When CER is as high as 1.0×10^{-3}, the advantage of the higher transmission rate can hardly be seen from the congestion window plot. Severe "oscillation" also affects the effective throughput achieved by TCP for he burst error model. However, the burst error model is affected more by the "multiple packet loss" phenomenon. Figure 5.12 illustrates that a burst of noise is more likely to destroy multiple packets when the transmission rate is high. The simulation results are plotted in Figure 5.14. Obviously there are more consecutive packet loss with the 8 Mbps rate than with the 1.5 Mbps rate.

When transmission rate increases, the bandwidth-delay product of the network increases. This means that the "pipe" of the network increases. However, the pipe cannot be filled as it grows because the steady-state average congestion window size achieved depends heavily on the packet loss probability. Figure 5.15 illustrates this phenomenon by plotting the increase of achievable throughput and the achieved

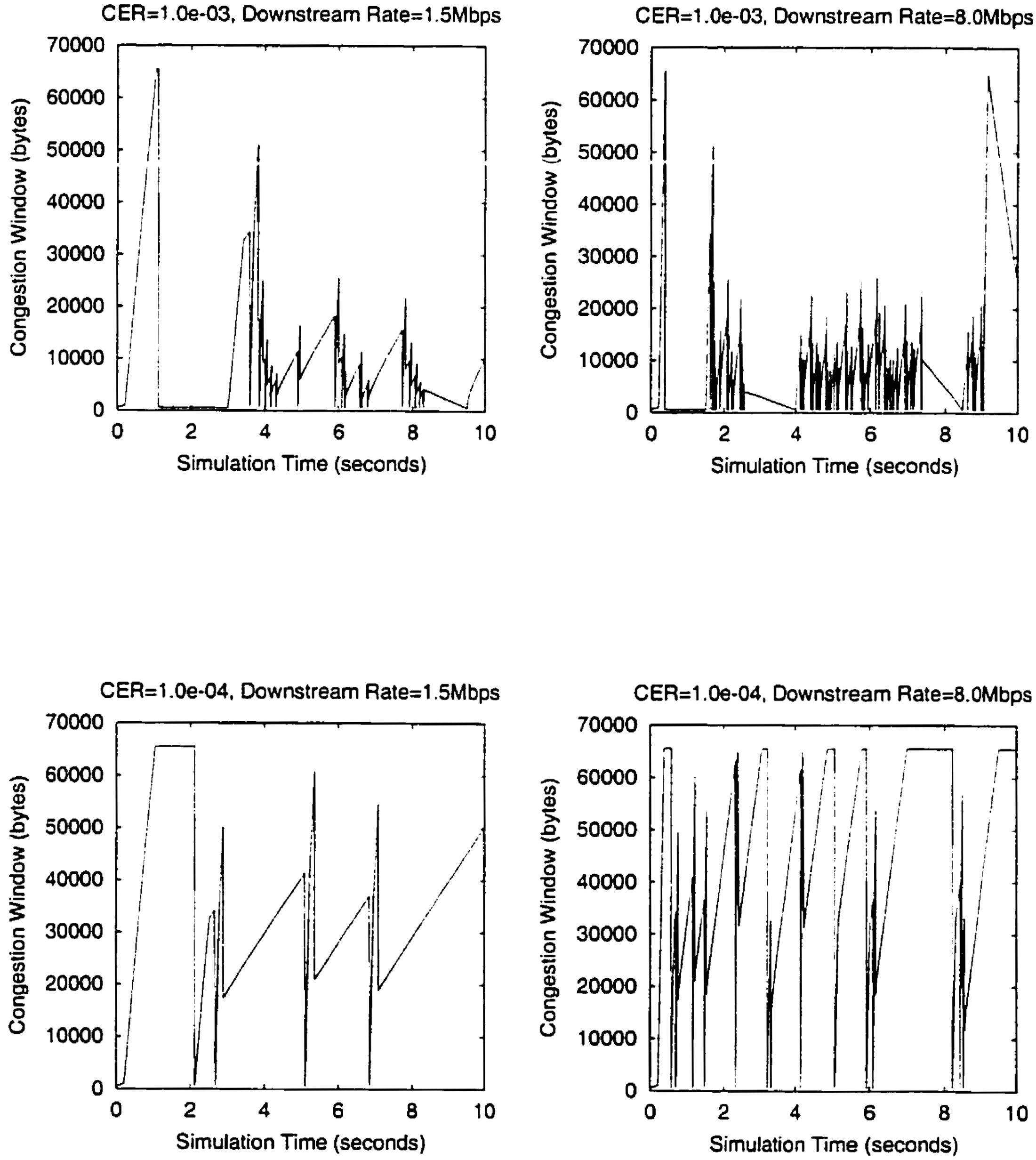


Figure 5.13: Congestion Window Illustration for Different Downstream Transmission Rates (Independent Error Model)

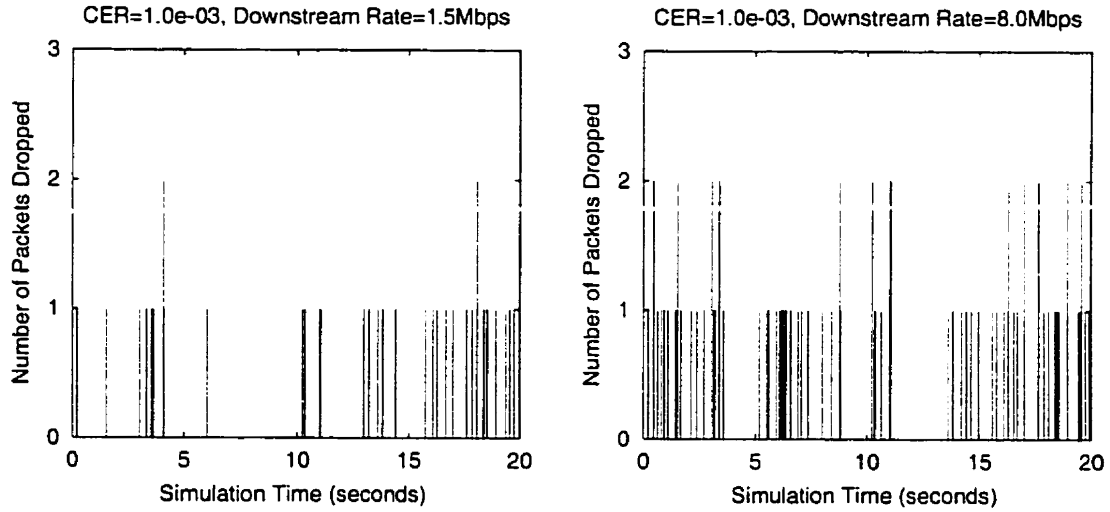


Figure 5.14: Packet Drops Illustration for Different Downstream Transmission Rates (Burst Error Model)

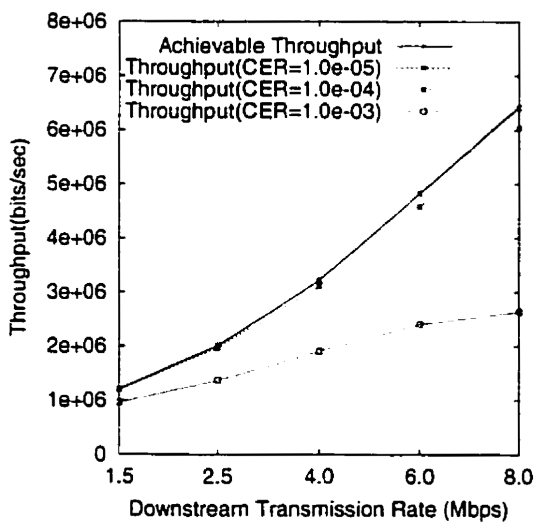


Figure 5.15: Comparison of Achievable Throughput and Achieved Throughput for Different Downstream Transmission Rates (Independent Error Model)

throughput along with the increase of transmission rates. When CER is high, the achieved throughput obviously lags behind the "ideal" throughput because of data loss by errors. It means that the growing speed of the denominator is higher than that of the numerator when calculating effective throughput (Equation 4.1). When the transmission rate is higher, the difference between the ideal throughput and the actual throughput becomes larger. This results in the decrease of effective throughput, especially when the transmission rate is high.

Experiments have also been done by varying the upstream transmission rates. Simulation results show that the change of upstream transmission rate has little effect on TCP performance. The main reason is that the upstream path only transmits ACKs in the experiments. It is seen that TCP is more sensitive to data loss than to ACK loss. It indicates that the downstream transmission rate is more important to the performance than the upstream transmission rate when downstream path transmits a large amount of data.

5.5 Effect of Some Noisy Lines

There are usually hundreds of thousands of ADSL lines at the customer premise within a real network. At a point of time, some of the lines may be noisy and cause data loss, while others may be in good status. The service providers want to know how many noisy lines they can tolerate on their networks without degrading performance. This experiment studies the relationship of a network's performance and the number of noisy lines within it. It was conducted with the eight TCP sources scenario. The factor is the percentage of noisy lines, which is the number of lines with errors divided by the total number of lines in the scenario. The ratio of noisy lines to total number of lines varies from 0% to 100% with an increment of 12.5%, and each noisy line has the same error ratio. Figure 5.16 and 5.17 illustrate how the change in percentage of noisy lines affects the performance. CLR, PLR, and effective throughput are calculated based on the statistics from all the lines (noisy or not) in the network scenario.

The CLR increases linearly along with the increase of noisy lines. An intuitive conjecture is that the ratio of CLR to CER is the same as the percentage of noisy lines. In order to prove the conjecture and to obtain a close view of the performance, the CLR values for both models from the simulation statistics are listed in Table 5.2. The data is not rounded so that the increase of CLR along with the increase of noisy line percentage can be observed more clearly. The calculated CLR in the table is based on the conjecture above. For example, it is expected that when the percentage of noisy lines is 12.5%, the CLR will be 12.5% of CER. Since CER equals 1.0×10^{-3} in this case, the expected CLR will be 0.000125. This is how the CLR values in column 2 are calculated. Measured CLR values are collected from the simulation results. Comparing the calculated CLR with the measured CLR, it is easy to find that the values are close. Simulation statistics for other CER values also show the same results although they are not shown here. The experimental results support the conjecture mentioned above.

The relationship of percentage of noisy lines, CER on the lines, and CLR of the network can be described as follows. If on a network there are n lines, the CER for each noisy line is e, and the percentage of noisy lines is p, then the overall CLR tends to be p percent of e. This observation implies that on a real network, if the

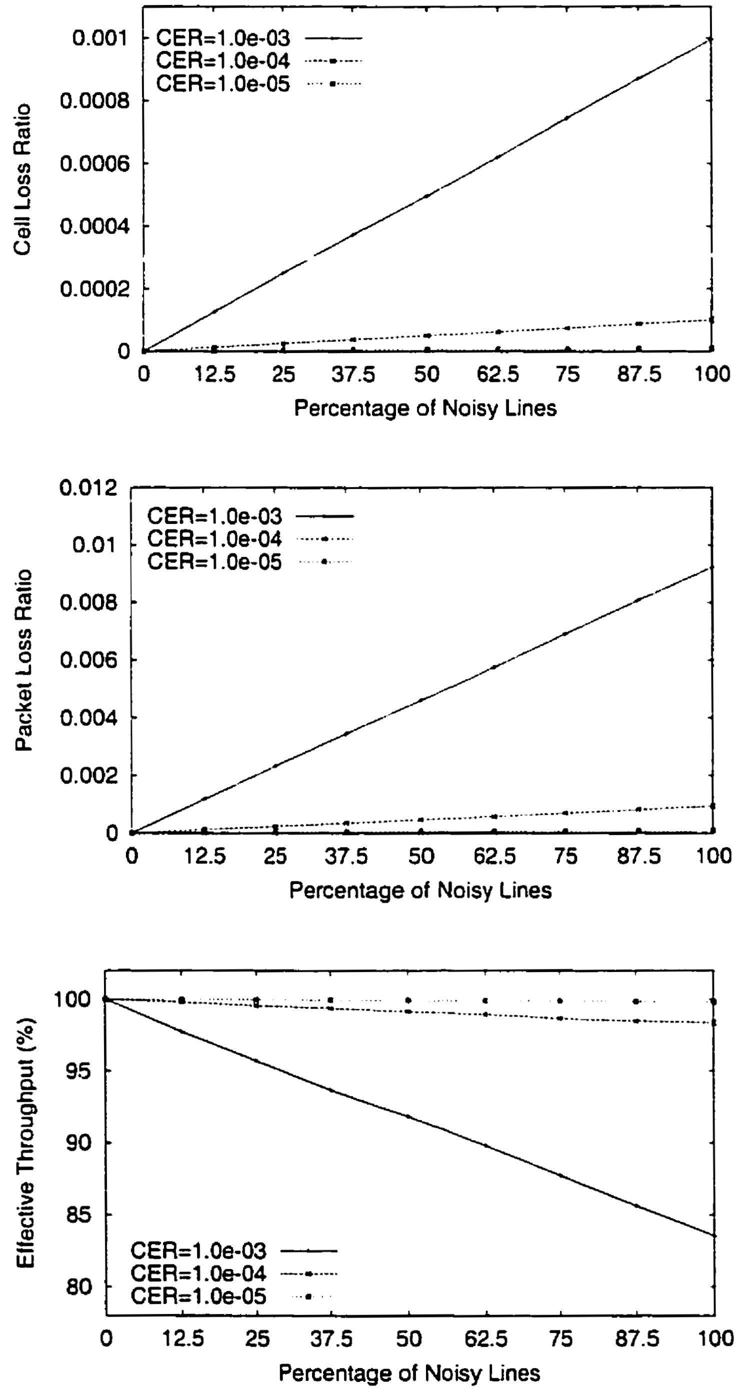


Figure 5.16: Effect of Percentage of Noisy Lines on CLR, PLR, and Effective Throughput (Burst Error Model)

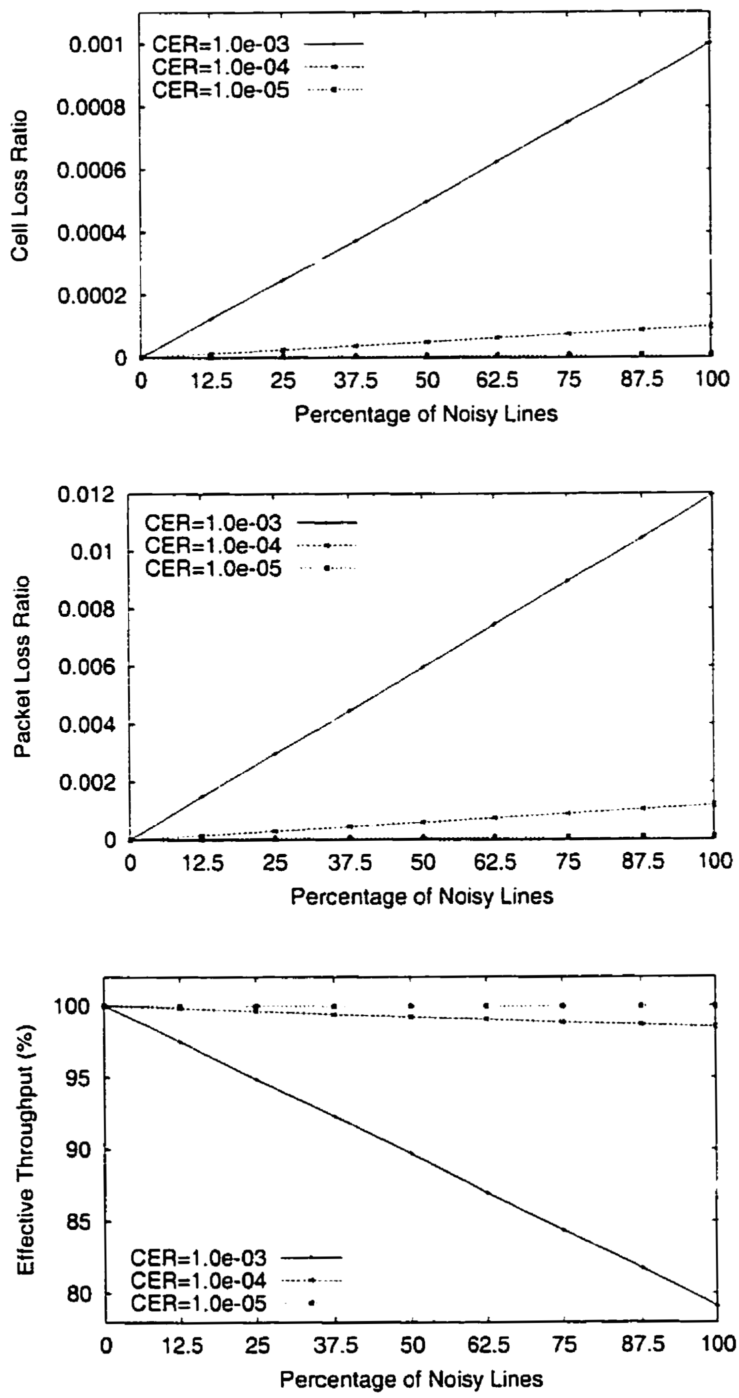


Figure 5.17: Effect of Percentage of Noisy Lines on CLR, PLR, and Effective Throughput (Independent Error Model)

Table 5.2: Comparison of Calculated and Measured CLR with a Different Percentage of Noisy Lines (CER = 1.0×10^{-3})

Percentage of Noisy Lines	Calculated CLR	Measured CLR	
		Burst Model	Independent Model
12.5%	0.000125	0.0001275	0.000125
25%	0.00025	0.000251625	0.000248625
37.5%	0.000375	0.000373875	0.0003735
50%	0.0005	0.000497125	0.000498625
62.5%	0.000625	0.000621375	0.00062475
75%	0.00075	0.0007465	0.000751125
87.5%	0.000875	0.000871125	0.000876125
100%	0.001	0.0009955	0.00100212

CER for each noisy line is known. the real error ratio on the network could be lower than this CER value because it is not likely that all the lines are noisy at the same time.

Similarly. the PLR and effective throughput plots also reveal the linear increment and decrement along with the increase of noisy lines. It is not a surprise because they are affected by CLR.

The above results are obtained using the network scenario shown in Figure 4.3. In this scenario. the transmission rate for the link between the ATM switch and ADSL Access Node is 155 Mbps (OC-3). Thus the ADSL lines in the local loop are the bottleneck links. When congestion occurs, queues are built up at the ADSL Access Node. Each ADSL line has a separate queue. Experiments have also been done with four lower rates for the link between the ATM switch and ADSL Access Node: 6 Mbps, 10 Mbps, 12 Mbps. and 14 Mbps. By varying this link rate, three things might be changed in the experiments: the bottleneck, the location and length of queues, and the buffering schemes (shared or independent). When the link rate is 6Mbps and 10Mbps, the bottleneck is moved from ADSL lines to this link. A large queue is built up at the output port of the ATM switch. The queue is shared by all the ADSL lines connected to this switch. Queues are still built up at the ADSL Access Node, though the length for each queue is smaller than that when ADSL lines are the bottlenecks. When the link rate is 14Mbps, the bottleneck is pushed to the ADSL lines again because 14Mbps is larger than the aggregate bandwidth of the eight ADSL lines in the scenario. The queue at the ATM switch side becomes shorter, and the queue for each ADSL line becomes longer.

Figure 5.18 compares the values of CLR, PLR and effective throughput with different link rates between the ATM switch and ADSL Access Node. Bar charts are used for easy comparison. There are five different values for comparison at each point. From left to right they represent the link rate of 6Mbps, 10Mbps, 12Mbps, 14Mbps, and 155Mbps. Link rate of 155Mbps is the scenario for Figure 5.16 and 5.17. The Cell Error Ratio (CER) is 1.0×10^{-4}, and the burst error model is used.

Compared with the results when the link rate is 155Mbps, the CLR and PLR plots also show a linear increment along with the increase of percentage of noisy lines using lower link rates. When the link rate is 6Mbps and 10Mbps, the decrease of effective throughput is subtle because the network is overloaded. When the link rate is 12Mbps, 14Mbps and 155Mbps, the effective throughput decreases linearly along with the increases of the percentage of noisy lines.

Figure 5.19 compares the fairness index for the five link rates. For all the link rates, the fairness index is higher when all the lines have similar condition (good

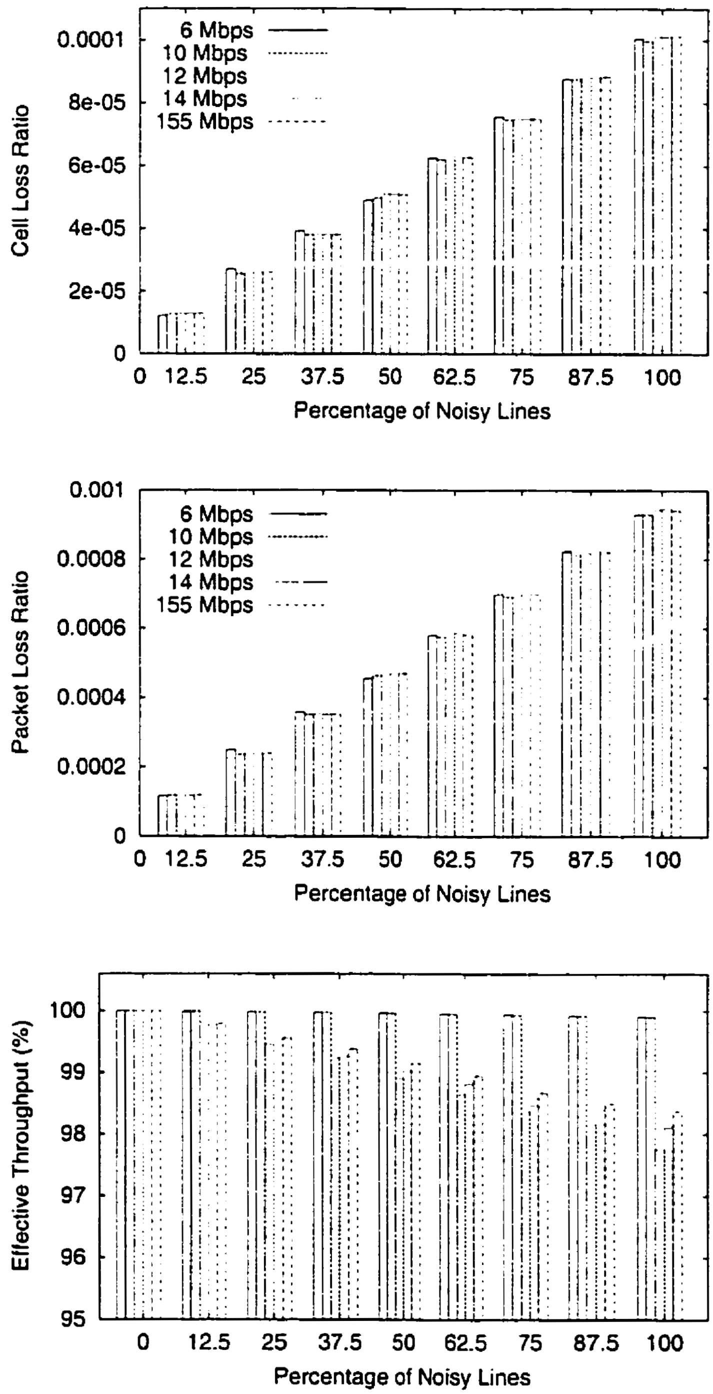


Figure 5.18: Effect of Percentage of Noisy Lines with Different Bottleneck Links (Burst Error Model, CER=1.0×10^{-4})

or noisy) and lower when only a portion of the lines are noisy. This trend can be seen clearly when the link rate is 6Mbps and 10Mbps, and is too subtle to be observed with other link rates. The decrease of fairness index with lower link rates is expected because shared buffer and congestion on the shared link aggravate the contention among sources.

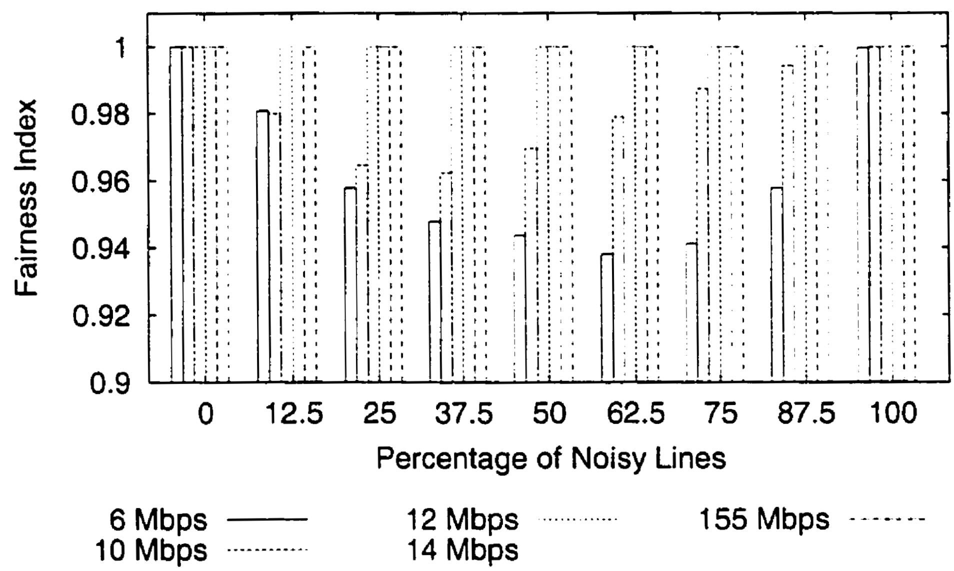


Figure 5.19: Comparison of Fairness Index with Different Bottleneck Links (Burst Error Model, CER=1.0 × 10^{-4})

5.6 Effect of Switch Buffer Size

In previous experiments, the switch buffer size is set large enough so that there is no cell loss due to congestion on the network. The purpose is to isolate the cause of cell loss to transmission errors (noise). However, errors and congestion usually exist concurrently on a network in the real world. This experiment is designed to study the network performance under the existence of both errors and buffer overflows.

Two factors are varied in the experiment: the switch buffer size and CER on the lines. Congestion occurs at the ADSL Access Node when servers send a large amount of data to clients from fast OC-3 links to low speed ADSL lines (Figure 4.3). Downstream cells are first queued and buffered at the ATM access switch within the ADSL Access Node before they reach the ADSL lines. If the buffer overflows, the arriving cells are dropped. Otherwise, the cells will be transmitted over ADSL lines to the clients. These cells are subject to errors on the ADSL lines and errored cells will be dropped by ADSL remote devices. Figure 5.20 illustrates the changes in PLR and effective throughput with different buffer sizes under three CER values. Figure 5.21 helps to explain the results by plotting cell drops due to errors and congestion separately.

When CER $= 1.0 \times 10^{-3}$, PLR and effective throughput seem to be not affected by the buffer size. It is because less data is transmitted due to the high error ratio. From the congestion window plots in Subsection 5.1.2 it is seen that the TCP congestion window can hardly grow because of frequent timeouts and slow starts. Therefore, the queue in the switch buffer is short. More data loss by congestion can be seen when the buffer size is smaller than 1000 cells in the experiment. The plots in Figure 5.21 show that cell drop due to errors is the dominant factor when CER $= 1.0 \times 10^{-3}$. In this case, though the requirement of buffer sizes is reduced, the TCP performance is poor because of errors.

When CER $= 1.0 \times 10^{-4}$, congestion becomes the dominant factor that affects TCP performance. The plots in Figure 5.21 show that the number of cells dropped by buffer overflows is much higher than cell drops by errors. When the buffer

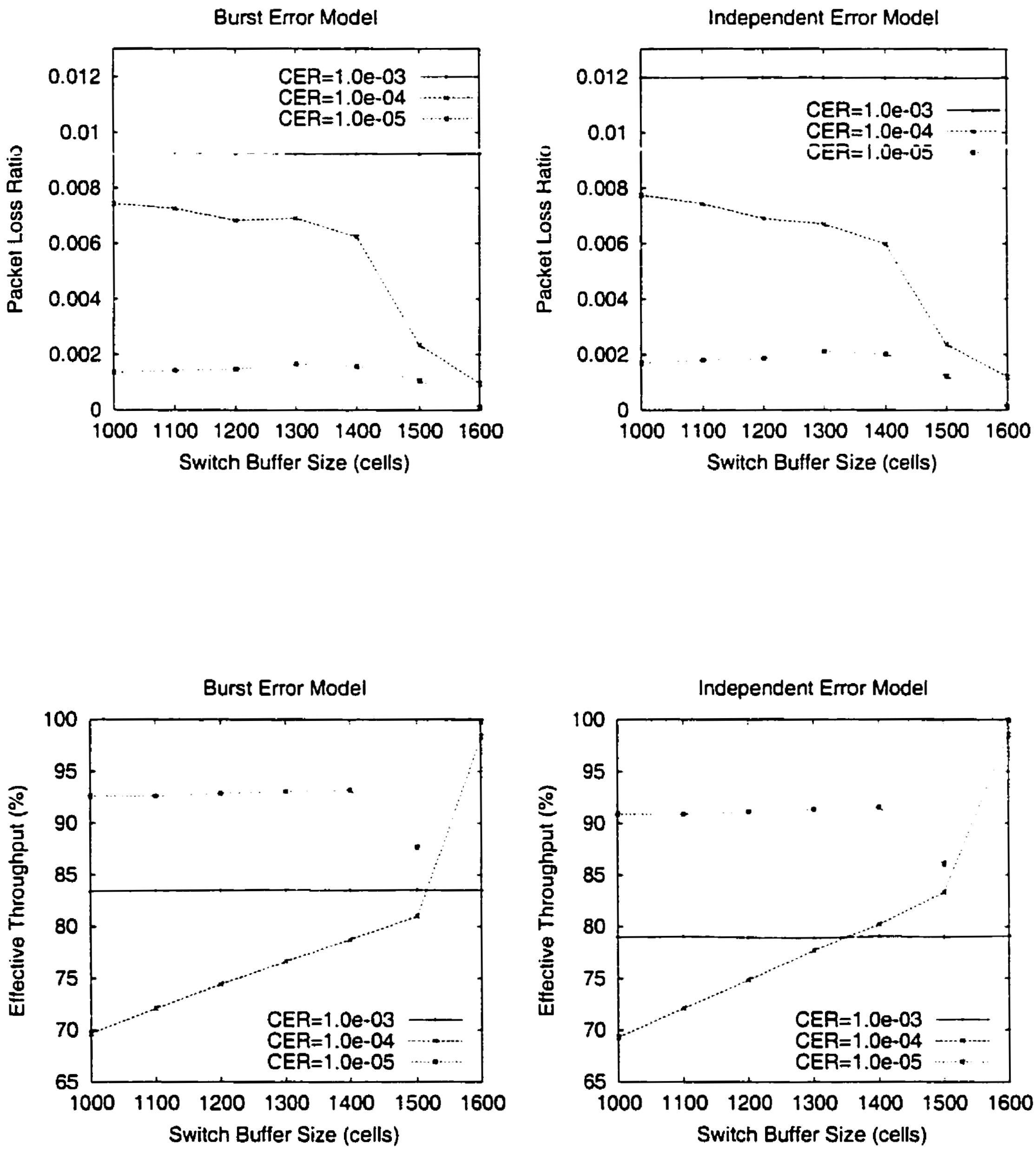


Figure 5.20: Effect of Switch Buffer Size on PLR and Effective Throughput

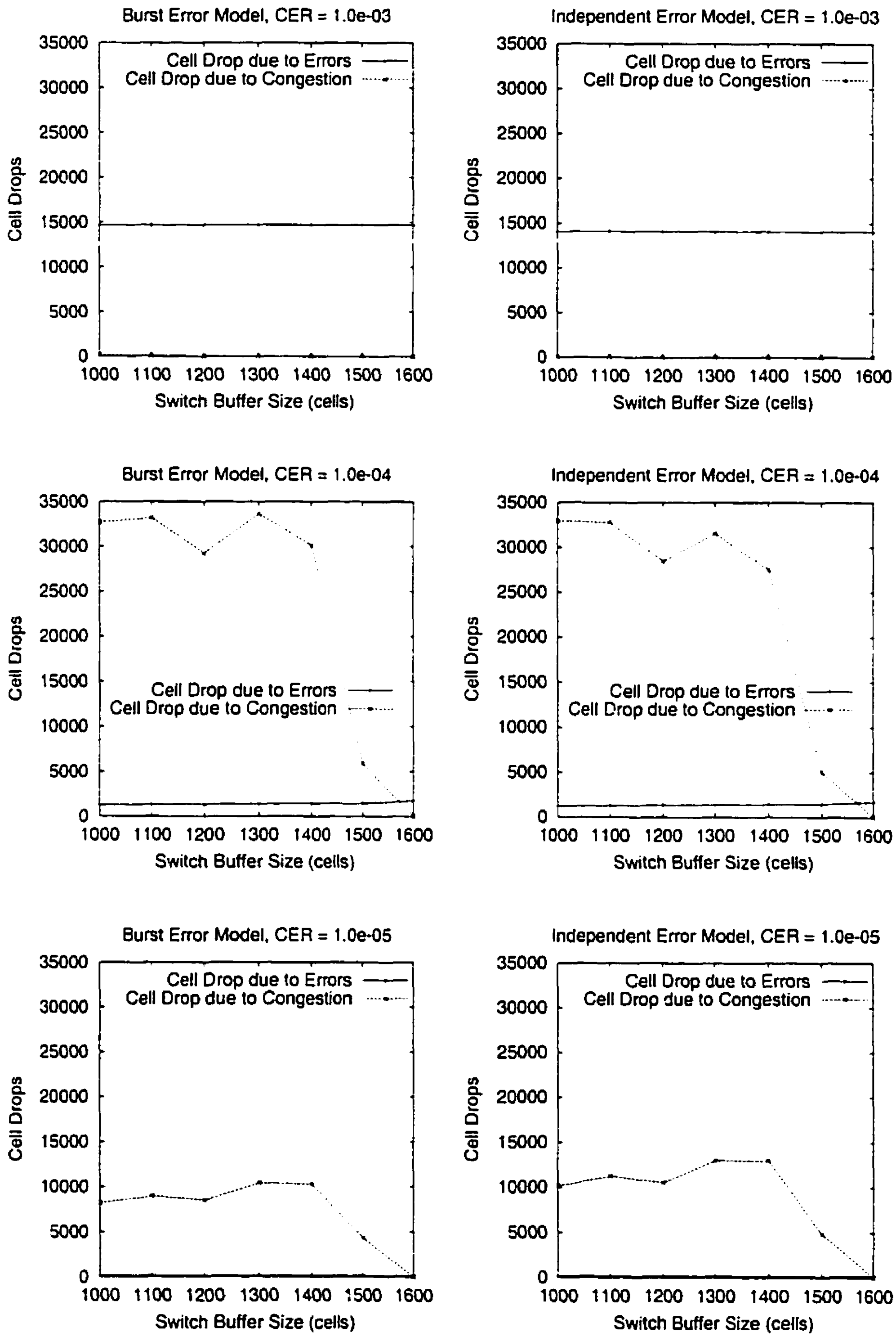


Figure 5.21: Cell Drops due to Errors and Congestion

size increases, PLR decreases and effective throughput increases. The effective throughput is sometimes even lower than that when CER $= 1.0 \times 10^{-3}$. It indicates that the large amount of consecutive packet loss due to buffer overflows is more harmful to TCP than the small amount of scattered packet loss due to errors.

When CER $= 1.0 \times 10^{-5}$, cell drops due to both noise and congestion decrease so that PLR is lower and effective throughput is higher compared with the results with higher CER values. This can be attributed to the lower CER. It is straightforward that fewer cells will be lost by noise with a lower CER value. The congestion is also alleviated with the lower CER because the number of retransmitted packets decreases. Another reason is that with a lower CER, fewer ACKs are lost. This enables TCP to react to data loss quickly.

By comparing the results above, it is seen that the requirement of buffer space varies when CER changes. When CER is low and noise is not the major factor for cell loss, the buffer size is decided by the bandwidth of bottleneck links and TCP sources. When CER is high, the large number of retransmitted packets worsens the congestion and requires a larger buffer size. When CER is extremely high and cell loss due to errors becomes the dominant factor, the need for buffer space reduces because less data is transmitted.

In Figure 5.20, there is a sharp drop of effective throughput when CER $=$ 1.0×10^{-5} and the buffer size is around 1500 cells. The phenomenon can be described as follows. The maximum queue length that can be generated in this scenario is about 1515 cells, based on the delay-bandwidth product and the maximum TCP window size for each connection. When the buffer size is larger than the maximum queue length, there is no overflow and the throughput is high. For example, buffer size = 1600 cells falls into this category and the effective throughput approaches 100%. When the buffer size is smaller than the maximum queue length, for example, buffer size = 1400 cells, TCP drops many cells each time congestion happens. The TCP sender reacts to the data loss by reducing the rate of transmitting. The queue in the switch buffer shrinks to a very small size and grows again slowly. The throughput decreases when the buffer size decreases. However, when the buffer size is smaller but very close to the maximum queue length (around 1500 cells in this scenario), TCP is forced into a syndrome of "frequent small amount of data loss". For example, when buffer size = 1514 cells, the queue built in the buffer is always close to the maximum buffer capacity, and TCP drops 2 cells every time when new cells come and fill the buffer. Since the amount of drop is small, the TCP sender can hardly perceive and react to the congestion on the network. It keeps on sending data at the rate it was prior to the congestion. Therefore, TCP is in a vicious circle of transmitting, dropping, and retransmitting data. This syndrome causes the sharp drop in throughput.

5.7 Observations

The experiments have explored and revealed some TCP features and its performance over ATM over lossy asymmetric networks. Some observations have been made based on the simulation results and summarized as follows.

- Loss due to noise can affect the end-to-end TCP performance severely when the error ratio is high (equal to or higher than 1.0×10^{-4} in these experiments). The experimental results consistently show that when Cell Error Ratio (CER) is high, the network experiences dramatic performance degradation, i.e., the sharp increase of packet loss ratio and decrease of effective throughput.

- Error distribution makes a difference. Scattered errors result in a higher packet loss ratio than burst errors. When the error ratio is low, a network with scattered errors achieves slightly higher throughput than a network with burst errors because TCP can easily recover from single packet loss using fast retransmit and fast recovery mechanisms. When the error ratio is high (higher than 1.0×10^{-4} in these experiments), TCP achieves higher throughput with burst errors than with independent errors.

- The structure of TCP over ATM inherently causes TCP performance degradation because of the size mismatch of TCP packets and ATM cells. Since the size of an ATM cell is fixed, the larger the TCP packet, the poorer the performance is, especially when CER is high.

- A line with a higher transmission rate is more vulnerable to noise. Under the same error ratio, the faster lines result in lower effective throughput (efficiency) than the slower lines.

- In an asymmetric structure, the performance is decided by the downstream transmission rate when the downstream path transmits a large amount of data, while the upstream path only transmits a small amount of requests.

- The aggregate error ratio on a network is subject to the percentage of noisy lines and the error ratio on each noisy line. If each noisy line has the same error ratio e, the overall error ratio on the network is lower than e when only some percentages of lines are noisy.

- The requirement of buffer sizes by noisy lines varies according to the error ratio. Generally more buffer space is needed because noise causes retransmissions.

- The maximum queue length forms the "critical point" for the switch buffer size. When the buffer size is larger than this point, there is no overflow and high throughput can be achieved. When the buffer size is very close but smaller than this point, the throughput decreases greatly because TCP falls into the syndrome of very frequent small data loss.

5.8 Summary

This chapter has presented and analyzed the experimental results produced by the simulation model of TCP over ATM over lossy asymmetric networks. The TCP performance under data loss by noise has been studied intensively. Simulation results show that TCP performance degrades dramatically when the error ratio on the network reaches a certain level. When error ratios are the same, different distributions of errors affect TCP performance to different degree. Other factors, such as large TCP packet sizes and high transmission rates can also make TCP more sensitive to errors. When only a part of the network is noisy, the decrease of network performance is proportional to the increase of percentage of noisy lines on the network. A very noisy network needs more buffer space because of the large amount of retransmissions caused by errors.

Chapter 6

Conclusions and Future Work

This chapter concludes the thesis with a summary of the work that has been done. The contributions of the thesis are listed. Future extensions of this work and research directions are also suggested.

6.1 Summary and Conclusions

The main objective of the thesis is to study the performance of TCP over ATM over lossy asymmetric networks (TCP/ATM/ADSL). A lossy network refers to a network with random errors caused by noise over the transmission lines. An asymmetric network refers to a network with different features in the downstream and upstream directions, for example, different bandwidth or delay. To be more specific, the lossy asymmetric network being studied in this thesis is the ADSL access network.

The method of doing this study is cell level simulation. An ADSL simulation model was designed based on the thesis objectives. It was implemented within an existing simulator called the ATM-TN, which was designed and developed by the Telesim project in the Department of Computer Science at the University of Calgary [27]. ATM-TN provides a way to study network performance at multiple protocol layers by simulating the ATM network structure and generating realistic traffic loads.

The ADSL simulation model includes the ADSL Access Node, ADSL Remote Devices and ADSL lines. The ADSL Access Node has the function of switching, multiplexing/demultiplexing, and error handling. It connects with multiple ADSL lines in the local loop. Each ADSL line has an ADSL remote device at the client site. The downstream traffic is demultiplexed by the ADSL Access Node and passed to the clients via noisy ADSL lines. Errors are detected by the ADSL remote devices and corrupt cells are dropped. The upstream traffic is multiplexed by the ADSL Access Node and passed to the backbone networks. Errors are detected by the ADSL Access Node and corrupt cells are dropped. The ADSL lines are error-prone copper pairs. The occurrence of errors on the ADSL lines has been simulated by two error models: the burst error model and the independent error model. Several experiments have been done to verify and validate the ADSL simulation model.

The goals of the thesis have been fulfilled by doing experiments with different network scenarios and analyzing the experimental results. Several groups of experiments have been designed and conducted using the simulation model. The performance of TCP/ATM/ADSL has been discussed and summarized in Section 5.7. It can be briefly concluded as follows:

- Loss due to noise affects the end-to-end TCP performance by causing packet loss ratio increase and throughput decrease. The impact is slight when the cell error ratio is low and becomes serious when the cell error ratio reaches a certain level. This suggests that the service providers can choose optimal values for good network performance.

- The scattered errors always cause a higher packet loss ratio than the burst errors. When the cell error ratio is low, the scattered errors have smaller effects on TCP than the burst errors because TCP can recover from single packet loss easily. When the cell error ratio is high, the burst errors have smaller effects on TCP than the scattered errors.

- Small TCP packet sizes (Maximum Segment Size) are favorable on a lossy network.

- Although more bandwidth in the local loop is the goal of ADSL, lines with higher transmission rates are more vulnerable to noise than slow connections.

- On an asymmetric network like ADSL, the downstream transmission rate is the key factor in the overall performance. The upstream rate only has subtle effects on the performance. Therefore, the goal should be the increase of downstream transmission rates and the improvement of transmission quality.

- The network performance changes linearly when a certain percentage of lines within a network are noisy.

- The change of TCP performance depends on the joint effects of congestion and noise, though congestion is usually the dominant factor. A noisy network requires more buffer space because of the retransmissions caused by errors.

The thesis has described how the objectives of the research are fulfilled. Chapter 1 provided a general introduction to the research background. Chapter 2 discussed the problems to study in depth by reviewing relevant research work and literature. Chapter 3 described the design and implementation of the ADSL simulation model and validated the model. Chapter 4 introduced the design of simulation experiments, the setup of network scenarios, and the performance metrics used. Chapter 5 presented and analyzed the simulation results. It also discussed the observations that have been drawn from the simulation results.

6.2 Thesis Contributions

This thesis has studied a network structure of TCP/ATM/ADSL on which research work has rarely been done. Besides the very limited literature in this area, previous studies of ADSL were mainly focused on the physical characteristics of the lines, not their effects on the higher protocol layers. This study simulated the noise characteristic of ADSL lines and made it the input factor to a full network stack. Therefore, this study examined TCP performance under a combined situation with packet segmentation/reassembly (TCP/ATM), noise, congestion, and asymmetry.

There is much work done on network congestion by other researchers, but relatively a little on the effect of noise. In this thesis, a detailed study on the effect of noise has been done. Several groups of experiments have been designed and conducted to explore the TCP performance over a lossy network. Some experiments are unique designs in this work, for example, the experiment on the percentage of noisy lines, and the experiment on both noise and congestion. Several observations based on the analysis of simulation results have been made suggesting possible ways to improve network performance.

This work has two advantages. One is the implementation of the burst error model and the comparison of its effects with the independent error model. It makes the simulation of errors more realistic. Other research work on lossy networks, such as [7], [24] and [25], only uses loss probability as the factor. They do not distinguish the features of different kinds of noise. Another advantage is the use of simulation as the study method. Some work only uses analytical models, like [24]. The ATM-TN used in this study is a high fidelity ATM network simulator. It provides an environment more similar to the real networks than is an analytical model.

An extensible simulation model has been established within the ATM-TN simulator. Further study can be performed based on this model with easy modifications and extensions.

6.3 Future Work

There are a lot of interesting areas in TCP/ATM/ADSL to be studied. Future work can be carried on in two directions. One is the enhancement and extension of the simulation model. Another is the improvement and enrichment of simulation experiments.

The simulation model can be enhanced in two ways. One is to extend the ADSL model to simulate more ADSL characteristics. Another way is to modify the ADSL model to simulate other networks with similar features, such as wireless networks.

The improvement and future work of experiments can be in two aspects. One is to use large network scenarios. Currently a scenario with eight TCP sources is used because it can explore the network performance clearly. However, it could be interesting to run some experiments with a larger scenario. Efforts have been made in the thesis work to use a large network scenario with more than a thousand ADSL lines, trying to simulate the capacity of a real multiplexer. Although restricted by the computer system resources and speed, a large scenario remains interesting to explore. Another aspect is to use the Web traffic model in addition to the TCP traffic model. The Web model is the simulation model at the application layer. It provides realistic traffic loads by simulating a user's behavior of Web browsing. Since ADSL is mainly for fast Internet access, it is meaningful to explore the Web model's performance over an ADSL access network.

Bibliography

[1] ADSL Forum. *ATM over ADSL Recommendation*, January 1999.

[2] M. Allman, V. Paxson, and W. Stevens. *TCP Congestion Control. IETF RFC 2581*. April 1999.

[3] ATM Forum. *Traffic Management Specification. Version 4.0*. April 1996.

[4] H. Balakrishnan, V.N. Padmanabhan, and R.H. Katz. The effects of asymmetry on TCP performance. In *MOBICOM 97*, pages 77–89. Budapest, Hungary, 1997. ACM.

[5] J. Banks and J. S. Carson. *Discrete-Event System Simulation*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1984.

[6] J. S. Chow, J. C. Tu, and J. M. Cioffi. A discrete multitone transceiver system for HDSL applications. *IEEE Journal on Selected Areas in Communication*, 9(6):895–908, August 1991.

[7] R. Cohen and S. Ramanathan. TCP for high performance in hybrid fiber coaxial broad-band access networks. *IEEE/ACM Transactions on Networking*, 6(1):15–29, February 1998.

[8] Paradyne Corporation. DSL sourcebook. http://www.paradyne.com/sourcebook_offer/sb_html.html.

[9] ADSL Forum. ADSL forum system reference model. tr-001. Technical report. ADSL Forum, 1997.

[10] R. M. Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):31–53, October 1990.

[11] P. Gburzynski, T. Ono-Tesfaye, and S. Ramaswamy. *ATM-TN Network Model Design*. WurcNet Inc., September 1995.

[12] F. Gomes, S. Franks, B. Unger, Z. Xiao, J. Cleary, and A. Covington. SimKit: A high performance logical process simulation class library in C++. In *Proceedings of the 1995 Winter Simulation Conference*, pages 706–713, Arlington, Virginia, 1995.

[13] F. Gomes and B. Unger. *ATM-TN System Design*. WurcNet Inc., September 1995.

[14] W. Goralski. *ADSL and DSL Technologies*. McGraw-Hill Companies, Inc., New York, NY, 1998.

[15] R.J. Gurski and C.L. Williamson. TCP over ATM: Simulation model and performance results. In *Proceedings of the 1996 IEEE International Phoenix Conference on Computers and Communications (IPCCC)*, pages 328–335, Phoenix, Arizona, 1996.

[16] ITU-T Recommendation I.321. *B-ISDN Protocol Reference Model and Its Application*, 1991.

[17] ITU-T Recommendation I.362. *B-ISDN ATM Adaptation Layer (AAL) Functional Description*, 1993.

[18] Information Sciences Institute, University of Southern California. *Transmission Control Protocol, IETF RFC 793*, September 1981.

[19] V. Jacobson. Congestion avoidance and control. *Computer Communication Review*, 18(4):314–329, August 1988.

[20] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, Digital Equipment Corporation, September 1984.

[21] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. Improving TCP throughput over two-way asymmetric links: Analysis and solutions. In *Proceedings of SIGMETRICS'98*, pages 78–89, Madison, WI, USA, 1998.

[22] M. Kara and M.A. Rahin. *Towards a Framework for Performance Evaluation of TCP Behavior over ATM Networks*. University of Leeds, School of Computer Studies Research Report Series, May 1997.

[23] N. A. Kheir. *Systems Modeling and Computer Simulation*. Marcel Dekker Inc., New York, 1988.

[24] A. Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4):485–498, August 1998.

[25] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, June 1997.

[26] R.O. Onvural. *Asynchronous Transfer Mode Networks Performance Issues*. Artech House, Inc., Norwood, MA, 1995.

[27] The Telesim Project. http://warp.cpsc.ucalgary.ca/TeleSim/.

[28] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. In *Proceedings of the 1994 ACM SIGCOMM Conference*, pages 79–88, London, England, 1994.

[29] G.C. Sackett and C.Y. Metz. *ATM and Multiprotocol Networking*. McGraw-Hill Companies, Inc., New York, NY, 1997.

[30] W. R. Stevens. *TCP/IP Illustrated*, volume 1. Addison-Wesley Publishing Company, Inc., 1994.

[31] W. R. Stevens. *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, IETF RFC 2001*, January 1997.

[32] A.S. Tanenbaum. *Computer Networks*. Prentice-Hall Inc., Upper Saddle River, NJ, third edition, 1996.

[33] B. Unger, F. Gomes, Z. Xiao, P. Gburzynski, T. Ono-Tesfaye, S. Ramaswamy, and C. Williamson. A high fidelity ATM traffic and network simulator. In *Proceedings of the 1995 Winter Simulation Conference*, pages 996–1003, Arlington, Virginia, 1995.

[34] R. van Melle, C. Williamson, and T. Harrison. Diagnosing a TCP/ATM performance problem: A case study. In *Proceedings of IEEE Globecom*, pages 1825–1831, Phoenix, Arizona, November 1997.

[35] C. Williamson, M. Arlitt, Y. Chen, and R. Gurski. *ATM-TN Traffic Model Design*. WurcNet Inc., September 1995.