Fiber-Based ATM Computer Network Performance and Survivability Issues Under Soft Failure Conditions

Igor Kostic

A Master's Thesis Submitted to the Faculty of the
Bradley Department of Electrical and Computer Engineering
of Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Ira Jacobs, Chair
Scott Midkiff
Rick Claus

12 August 1999
Blacksburg, Virginia

Keywords: Soft Failures, ATM, Optical Fiber, Computer Networks

Fiber-Based ATM Computer Network Performance and Survivability Issues Under Soft Failure Conditions

Igor Kostic

(ABSTRACT)

In this research a suspected “gray” area in computer network performance that might exist between the network’s optimum performance and complete network failure was investigated. Changes in performance were studied as seen by the application layer of the network as attenuation was introduced at the physical layer of the network. The network performance was measured for four groups of tests running over TCP and UDP. Various scripts emulating different network applications were used. Ultimately the performance was evaluated at different optical power levels, based on throughput, transaction rate, response time, and lost data. Additionally, the bit error rate characterization of the attenuated optical fiber link was analyzed. Ultimately, performance of the fiber optic ATM-based computer network as seen by the user was correlated with optical power degradation introduced in the physical layer. A decrease in performance was found as attenuation was increased, even in the range of “normal” operating conditions. As expected, TCP, although not immune to soft failures, performed better than UDP. A description of the investigation, tests, scripts, methods and results are included.

Acknowledgements

Robert Monine and Roger Hawk of Olicom, for the use of, and expertise on Olicom ATM switches and Olicom ATM Network Interface Cards.

Robert Martin, Steve Engemen, Bill Lindblom and Ron Montgomery of Force, Inc., for generous help with BER measurements, optical power meters and variable attenuators.

Warren Stuart, Leo Guzynski, Julie Cantando, and Ann-Christin M. Pautz of Ganymede Software, Inc. for assistance and guidance with Chariot Software.

Tom Paredes and Jim Bandy of Litton Network Access Systems for help with Video Codecs and ATM switches.

Dr. Ira Jacobs, for being a great, extremely patient and supportive advisor. In addition, his acquisition of the PCs was essential to this project.

Dr. Scott Midkiff, for his expertise on computer networks.

Dr. Rick Claus for giving me the chance, opportunity, and means to pull this project all together

Robin Rogers for unbelievable patience and support as well as invaluable help with putting the manuscript together.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In recent years the world has witnessed exploding demands on communication networks. In addition to more conventional communication devices such as the telephone and television, the development of the Internet provided for interconnectivity of personal computers. This interconnectivity enabled a transition of the personal computer from computing device into a new kind of personal communication device, ultimately creating an enormous world-wide network of these PCs. The booming trend of the Internet has overwhelmed the existing communication infrastructure and has placed demands non-existent just a decade ago.

To meet some of these demands, there has been a global effort to merge the segregated networks into one integrated single network capable of merging long standing foes -- data, voice and video -- into a Broadband Integrated Services Digital Network (B-ISDN). Prior to the integration, data was generally transported by separate and relatively slow packet-switched networks over existing telephone lines; voice was transported via circuit-switched networks; and video was broadcast over separate cable networks. The integration of these services into one network that delivers all of the disparate services will increase the overall efficiency of networks, as well as raise the potential for higher revenues, thus creating a greater ability to meet the growing customer demand for these services.

However, the same integration will make these networks operate at higher risk, since the amount as well as the importance of information flowing through will increase. This condition, in turn, will increase the importance of the networks to offer reliable and continuous performance. Furthermore, just offering a capability to transfer data is becoming an insufficient condition for good performance of communication networks; the birth of numerous new services such as e-commerce and video-conferencing have further expanded requirements where the quality of the service offered is becoming just as important as the service itself.

In other words, it is not only important that the data can get from point A to point B, but also how fast and with what quality it can get there. Therefore, anything that impairs the network's capability to offer a service, as well as the quality of that service, also plays an extremely important part in the overall trend of analyzing and retrofitting today's networks and planning, and designing future networks to make them more immune to any kind of failure.

The escalating demands also have resulted in much work on categorizing, classifying, and analyzing network failures and, although somewhat ambiguous, classification of failures has received a lot of attention from the research community and telecommunications standards-setting bodies. The classification has been focused primarily on complete or hard failures of networks or their parts, looking at whether a service is offered; it has basically not addressed the possibilities of the effects of soft failures on network performance which could potentially affect the quality of an existing service.

Generally, soft failures include a very broad range of conditions, which do not completely terminate a link, connection or a service. Rather, in a subtle way, they impair and reduce the overall performance of a network. Due to overwhelming end user demand for quality, maximum bandwidth and speed of connectivity, it is of paramount importance that existing as well as future networks perform at optimum levels. Even more important than extracting the maximum bandwidth out of the infrastructure is the influence soft failures can have on BER, which in turn may wreak havoc on quality of service (QoS). Under these strenuous conditions, the possibility of “slight” performance decreases potentially caused by soft failures is becoming an important factor.

Well-designed, installed, and operated fiber optic networks should not, in principle, experience soft failures. In real networks, however, there is the potential for soft failures such as unexpected or unaccounted attenuation from microbends and cracks, and improper fiber and / or connector use resulting from improper installation. It is expected that the horizontal cabling associated with fiber to the desktop will be much more exposed and thus vulnerable to a variety of soft failures. Furthermore, the multilayered nature of computer network protocols only adds new possibilities for creating different kinds of soft failures, as well as for hiding and masking the existing ones.

This research investigates the suspected “gray area” of network performance that lies somewhere between optimal performance and complete failure, and thought to be affected by such soft failures as mentioned above. To determine the existence of such an area, soft failures are emulated in a laboratory network setup by inserting variable attenuation in the fiber connecting two ATM switches. By applying gradual attenuation to mimic soft failures, and then using a powerful software tool to generate and monitor network traffic, it is possible to detect deterioration or degradation of network performance.

1.1. Research Objective

As mentioned before, a great deal of previous research has been done in classifying failures. Based on the work of ANSI Technical Subcommittee T1A1, three features -- unservability (U), duration (D), and weight (W) – can be used to describe network failures as catastrophic, major or minor [1]. Catastrophic network failures are often triggered by such disastrous occurrences as earthquakes, floods, tornadoes, power grid failures, or war. Major cable backbone failures can occur as a result of fire, terrorism, or tandem switch failure. Minor single link or path failures often are due to fiber failure, equipment failure, and shelf or unit failure.

Most of the recovery solutions for these failures have been incorporated in the lower layers of networks such as SONET and ATM, and usually include individual layer restoration features such as automatic protection switch (APS), self-healing ring (SHR), self-healing network (SHN), or failure-resistant virtual path (FRVP).

The multilayered nature of computer networks (for example, WDM/SONET/ATM/IP) makes the problem of network restoration more complex; nevertheless, recovery solutions have also been offered such as layer escalation, subnetwork escalation, and scheme escalation.

Soft failures are not as easy to define and have not yet been studied as extensively as hard failures. Most of the higher layer protocols assume that the fiber is a perfect medium and that errors are either the fault of other network malfunctions or there is a complete failure of the physical layer. To further complicate soft failure situations, if a certain level of degradation does exist in the fiber, such issues as congestion, contention, number of acknowledgements and retransmissions, proprietary equipment and code incompatibility make it even more difficult to determine the actual cause for a decrease in performance.

In addition, there is no “standard” network architecture – networks are all configured differently. As a result, it would be too impractical, maybe virtually impossible, to examine soft failures for all possible existing networks. Therefore, the goal of this thesis project is to provide insights into network performance under certain specific soft failure conditions with very controlled system parameters, rather than accomplish a conclusive study that is applicable to all networks. The hope is that after gaining insights into degradation versus performance, certain conclusions can be extrapolated to network performance in general.

To achieve this, the scope and size of the testbed network was limited, and real hardware and software were used for the setup, as opposed to employing a simulated network. It was determined that a “real” system would elicit better responses and would result in more useful, specific data.

The test set-up included two ATM switches (Olicom CrossFire 9100 and 9200), two PCs (450 MHz Pentium III processors with 128 MB of RAM), and two NICs (RapidFire 6162 ATM 155 PCI), all connected by OC-3 multimode optical fiber links. Furthermore, two variable attenuators were inserted between the two switches and the same set of tests was executed for different levels of attenuation.

The ATM switches were configured to run LANE, where the 9200 operated as the primary LANE administrator and the 9100 was the secondary. The emulated protocol was Ethernet (IEEE 802.3). This configuration offered a SONET / ATM / LANE / Emulated Ethernet / IP / TCP / Application protocol stack.

The application layer scripts and network performance tests were performed using Ganymede’s Chariot software [2]. Chariot is a software tool designed to test end-to-end network performance of complex and multiprotocol networks, and is distributed on the PCs as end users of the network.

For this project, four main tests were selected; three emulated Bader[1] benchmark or classic transactions, internet applications, and multimedia data; the fourth was a test designed for “stressing” the network by running 16 simultaneous connections between the two end points. Then, each test was repeated for increased attenuation levels until the ports on the ATM modules displayed a red light as an indication of insufficient optical power to the switch.

Performance is evaluated based on the following parameters which are observed and compared for different attenuation levels: throughput, transaction rate, and response time for TCP-based applications (Tests 1 and 2), and throughput, lost data and percent bytes lost for UDP-based applications (Tests 3 and 4).

1.2. Thesis Organization

Chapter 2 is a brief history of computer networks, and includes descriptions of the standardization initiatives ISDN and B-ISDN.

Chapter 3 provides an overview of computer networks, including topics such as OSI, SONET/SDH, and ATM. Bandwidth management of ATM networks is also described, as well as insights into different service categories offered by ATM, quality of service and flow control issues.

Chapter 4 gives a description of network failures, their classifications, network restoration techniques and "soft" failures.

[1] Chariot names this class of application scripts after Lance Bader who initially specified them for IBM’s System Network Architecture (SNA). Now, they can generally be applied to all network architectures.

Chapter 5 offers an explanation of the application layer of the computer network. It describes the scripts used for generating network traffic, as well as the metrics used for network performance evaluation.

Chapter 6 describes the configuration of the experimental setup, as well as the actual equipment used. It further explains the parameters of each component in the setup, and the response variables on which the evaluation model is based, such as attenuation, throughput, transaction rate, response time and lost data.

Chapter 7 presents the performance test results. It offers the obtained data for individual response variables, as well as the correlation between the network performance and the attenuation. It also attempts to relate the given attenuation through the fiber with the corresponding bit error rate.

Chapter 8 gives the conclusions and recommendations for further research.

Chapter 2. Historical Highlights and Standardization Initiatives

2.1. Historical Developments of Communication Networks

Incredible advances and developments of individual and service-specific communications networks characterize telecommunications of the 20th century. In addition to the creation of separate and sophisticated networks, the close of the century has also witnessed a push toward integration of all these separate communication environments into a single network -- one that would offer higher capabilities for meeting the needs of newly emerging services while still offering the same or even improved quality of the existing services at a lower cost. The historical developments leading to the need of networks integration are presented in the following sections.

2.1.1. Telephone Networks

At the beginning of the telecommunications industry, copper wire was quickly accepted as a standard transmission medium, mostly due to its good conductivity and low production cost. However, it quickly became obvious that a single message or logical link per physical copper link was insufficient and economically nonviable. Ways of packing multiple channels onto one physical link emerged and the resulting system is referred to as the carrier system, or simply the carrier [3]. These carrier systems can be both analog as well as digital, and both are in use today. The analog carrier systems, however, are decreasing in importance.

In the 60's the telephone networks began their transition from analog circuit switched networks to all digital circuit switched networks. The first digital carrier system introduced in public switched networks was known as the T-1 carrier system, and was designed to operate over twisted copper wire pairs [3]. The rates and parameters of T-1 carrier reflect the fact that it was designed for carrying voice. The basic building rate for T-1 carrier was derived from a bandlimited analog voice signal sampled at the rate of 8000 samples per second and then 8 bit pulse code modulated to give a digital signal at the rate of 64 kbps. T-1 signal carries 24 simultaneous voice channels for the total capacity of 1.544 Mbps. Four T-1 signals can be time division multiplexed to obtain a 96 channel 6.312 Mbps T-2 signal and seven T2 signals can be time division multiplexed to obtain a 672 channel 44.736 Mbps T3 signal [3].

In more recent years telephone networks have been used for other applications, such as facsimile transmission, and the resulting increased network usage have forced the transition of main backbones of the telephone networks from copper and microwave radio to optical fiber. Also, using pre-existing networks for new applications lead to certain shortcomings. The T-carriers, for example, perform well in the circuit switched public telephone networks for carrying voice, but the constant bit rate channels are proving to be less efficient for the transfer of variable bit rate data traffic which is becoming a significant percentage of all telecommunications needs.

2.1.2. Cable Television Networks

Television, in addition to radio broadcast, is transmitted via broadcast networks which until recently were generally based on the use of coaxial cable as the transmission medium. Although cable networks are a more recent development than telephone networks, they have been very slow in transitioning from analog to a digital format. Due to the broadcast nature of TV networks and widely accepted analog transmission, there has never been a need to standardize certain "carriers," although channel assignment for both over-the-air and cable transmission are standardized. The common method of simultaneously packing individual 6 MHz TV channels into one cable uses the subcarrier-multiplexing (SCM) technique of amplitude modulated vestigial sidebands (AM-VSB) TV signals.

Inherently large bandwidths of video signals have also pushed cable television (CATV) distributors to include fiber and hybrid fiber-coax as transmission media of choice. The larger bandwidth of CATV networks makes them lucrative for transmission of data, but the broadcast nature of the networks make any transition to support new services a difficult option, although there is considerable work directed towards this.

2.1.3. Data Networks

At first computers were created for the sole purpose of aiding humans in numeric calculations and computing. As their processing speeds improved, the amount of calculations performed increased drastically. Large amounts of processing required large data bases and large storage files, and soon single computing machines were not adequate for performing all the desired operations while simultaneously storing all the desired information. Soon there was a need for file sharing, parallel processing and distributed computing; all of which require fast and reliable ways of transporting the data between the individual computers.

Faced with this problem throughout the 60's, a number of people began thinking about connecting computers to each other to improve performance, efficiency and overall capabilities for computing. However, it wasn't until the annual meeting in 1967 of the "principal investigators" of the Advanced Research Projects Agency (ARPA), an R&D agency funded by the U.S. Department of Defense, that the topic of networking was brought up and seriously considered as a future project [4].

The ARPANET Completion Report, as published jointly by BBN of Cambridge, Mass. and ARPA states the following from the meeting:

"At the meeting it was agreed that work could begin on the conventions to be used for exchanging messages between any pair of computers in the proposed network, and also on consideration of the kinds of communications lines and data sets to be used. In particular, it was decided that the inter-host communication 'protocol' would include conventions for character and block transmission, error checking and retransmission, and computer and user identification. Frank Westervelt, then of the University of Michigan, was picked to write a position paper on these areas of communication, an ad hoc 'Communication Group' was selected from among the institutions represented, and a meeting of the group scheduled" (ARPA draft, III-26) [4].

ARPA's Program Plan for the ARPANET was titled "Resource Sharing Computer Networks," and it was approved by the Director in June 21, 1968, the same month it was submitted [4].

ARPANET was created in 1971 as the first packet-switched network, connecting military and civilian locations as well as some universities [5]. This data network was the beginning of a major evolution of computer networking, eventually giving birth to today's Internet.

Along with the increased connectivity, interoperability and decreased size and price of computers, the Internet helped promote the transition from the use of computers as primarily computational machines to their implementation as a new form of personal device. However, David Clark observed, "It is not proper to think of networks as connecting computers. Rather, they connect people using computers to mediate. The great success of the Internet is not technical, but in human impact. Electronic mail may not be a wonderful advance in Computer Science, but it is a whole new way for people to communicate. The continued growth of the Internet is a technical challenge to all of us, but we must never loose sight of where we came from, the great change we have worked on the larger computer community, and the great potential we have for future change [4]."

The arrival of affordable Personal Computers (PCs), coupled with the recent creation of the World Wide Web, facilitated explosive growth of the Internet (Figure 2.1). According to an article in Nature dated 8 July 1999, the Internet doubled in the past year, causing the number of web pages to escalate to 800 million today [6, 7].

There was quickly a need for more bandwidth, and a switch to higher capacity optical fiber as the data transmission medium was inevitable, just as it was necessary for the telephone and cable TV networks.

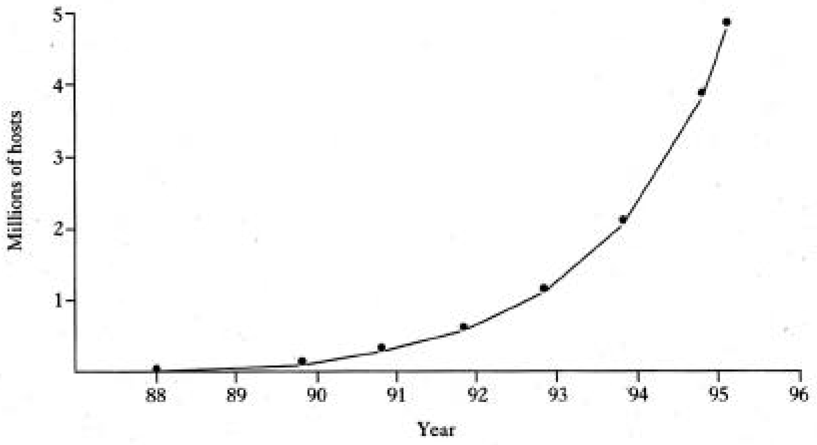


Figure 2.1 Growth in Number of Computers Connected to the Internet Since 1988 [6].

2.2. Standardization Initiatives

The transition of the three major communication environments (voice, video, data) to digital technology and the choice of optical fiber as the physical medium of choice for high speed communications in the wide area networks (WANs) exposed the issues of compatibility and interoperability as the main problems confronting global integration of these networks. At the beginning of commercial use of digital systems incorporating optical fibers as the transmission channel, there was no compatibility between vendors with regard to transporting signals over an optical network. Each vendor of optical transmission equipment, by necessity, produced proprietary signals. This problem was not limited to the U.S., and as other developed countries were running into similar problems, standardization of digital communication systems was becoming a global concern.

Two main avenues needed major work, and continue to require refinements, by the international standards setting community; first, it is necessary to implement a standardized method of transmitting the data no matter where it originates; second, a standard of procedures, rules and protocols needed to be developed. The adoption of both standards ensures the more efficient use of transmitted data by all geographically separated users.

The standardization initiatives concerning the integration of the disparate networks began with the adoption of concepts of Integrated Services Digital Network (ISDN) and Broadband-ISDN (B-ISDN), as well as the introduction of various protocol reference models such as the Open System Interconnection (OSI).

2.2.1. ISDN

Although the work on ISDN started in the mid 1970’s, the first official standardization move toward integration of networks carrying different types of traffic came in 1984 [8]. That year, the adoption of the I series recommendations concerning ISDN was recommended by the Plenary Assembly of the Comité Consultatif International Télégraphique et Téléphonique (CCITT) which is now known as the International Telecommunication Union (ITU) [9]. (ITU-T is a sector of ITU, which is in charge of setting network standards for public telecommunications.) The goal of these recommendations was to provide a standardized set of multipurpose user-to-network interfaces, which would provide an end-to-end connectivity among users, as well as access to many voice and non-voice digital services [9]. The ISDN’s building block rate of 64 kbps was derived in almost identical fashion to the T1 carrier for telephone networks, and the upper limit for ISDN channel rates was established at about 2 Mbps.

2.2.2. B-ISDN

Although the ISDN rates for network access were generally enough to support transporting voice and some lower grade video, they certainly were not enough for some multimedia applications such as high-resolution images and real-time video, which can require substantially higher rates. In addition, the ISDN rates are insufficient for connecting a LAN.

Thus, the CCITT/ITU-T began its work on extending the ISDN capabilities as early as 1985 [8]. This work eventually resulted in defining recommendations for B-ISDN by adding new high-speed channels to the existing ISDN rates, as well as new broadband user-to-network interfaces. However, the most important results to come out of this work were the establishment of new standards of synchronous techniques for transmission and asynchronous techniques for switching and multiplexing of broadband data services, now known as SONET and ATM.

Chapter 3. Overview of Computer Networks

3.1. OSI Reference Model

The International Standards Organization (ISO) was among the first organizations to attempt to define a standard for global communications between various computer systems. Its seven-layer reference model (Figure 3.1) was the basis for the Open System Interconnection (OSI) architecture that separates each functional area within any computer into discrete layers. This effectively simplifies the complex processes involved in communication between or among various cooperating computer systems [10]. Even though this OSI method provides a sensible conceptual framework for independent computer networks, the ISO protocols themselves have been a commercial failure [6].

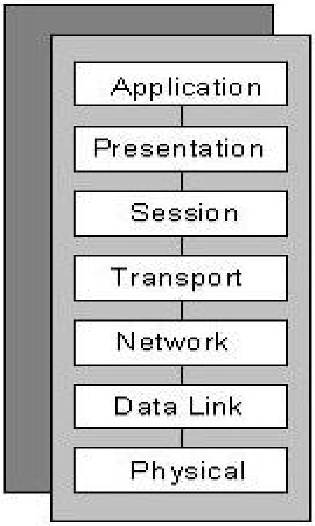


Figure 3.1 ISO’s OSI Reference Model

3.2. SONET/SDH

Synchronous Optical NETwork (SONET) and Synchronous Digital Hierarchy (SDH) represent closely related sets of standards that govern interface parameters such as rates, formats, multiplexing methods, and Operations Administrations, Maintenance and Provisioning (OAM&P) for high speed optical transmission [11]. SONET is becoming the primary set of standards used in North America, while SDH is primarily used in Europe and Asia.

From a transmission perspective they both provide an adequate international basis for existing time division multiplexing (TDM) as well as the new (cell-multiplexed) services. Initially, the goal of these standards was to facilitate the interworking of multivendor equipment across a single fiber span. This would allow attachment of different vendors' equipment without loss of functionality of the overall system. Furthermore, the goal was to provide for future increases in data rates by defining the base signal rate, which could then be synchronously multiplexed to attain higher rates.

SONET represents a basic physical layer technology that can carry any type of payload, both isochronous (delay-sensitive) voice and video, or packet switched data. It is a transmission technology, which lies at the bottom of the Open System Interconnection (OSI) stack. Above SONET, it is possible to implement any Data Link technology, including Asynchronous Transfer Mode (ATM), Frame Relay, Switched Multi-megabit Data Service (SMDS), and even FDDI [9].

Both SONET and SDH make use of the basic building block signals. For SONET the first level, or the smallest building block, is a Synchronous Transport Signal Level-1 (STS-1) with a signal rate of 51.84 Mbps. STS-1 corresponds to

Optical Carrier Level –1 (OC-1) signal in SONET systems. SDH starts at the base rate of 155.52 Mbps, which is the rate of the fundamental building block, referred to as Synchronous Transport Module Level-1 (STM-1). Thus an STM-1 level in SDH corresponds to an OC-3 level in SONET. Since the rate of STM-1 is just 3 times the STS-1, once at the STM-1/STS-3 level and higher, SONET and SDH are completely interoperable. (Table 3.1)

Table 3.1 SONET Transmission Rates

SONET Signal	**SDH Signal**	**Transmission Rate [Mb/s]**
OC-1		51.84
OC-3	STM-1	155.52
OC-12	STM-4	622.08
OC-24		1244.16
OC-48	STM-16	2488.32

The transmission of data over SONET/SDH lines occurs in frames, which are depicted as rectangular octet-based units. Due to the compatibility issues, both systems send frames 8000 times per second, or one every 125 µs. Both frames have very similar structure except for their dimensions, which reflect the previously mentioned differences in basic building block rates. SONET STS-1 frame format is 9 rows by 90 byte columns. Keeping in mind that 1 byte equals 8 bits, then the frame consists of a total of 6480 bits. (9 x 90bytes x 8bits/byte). Since 8000 such frames are sent per second, then the aggregate rate is 51.84 Mbps. (6480 bits/frame x 8000frames/s).

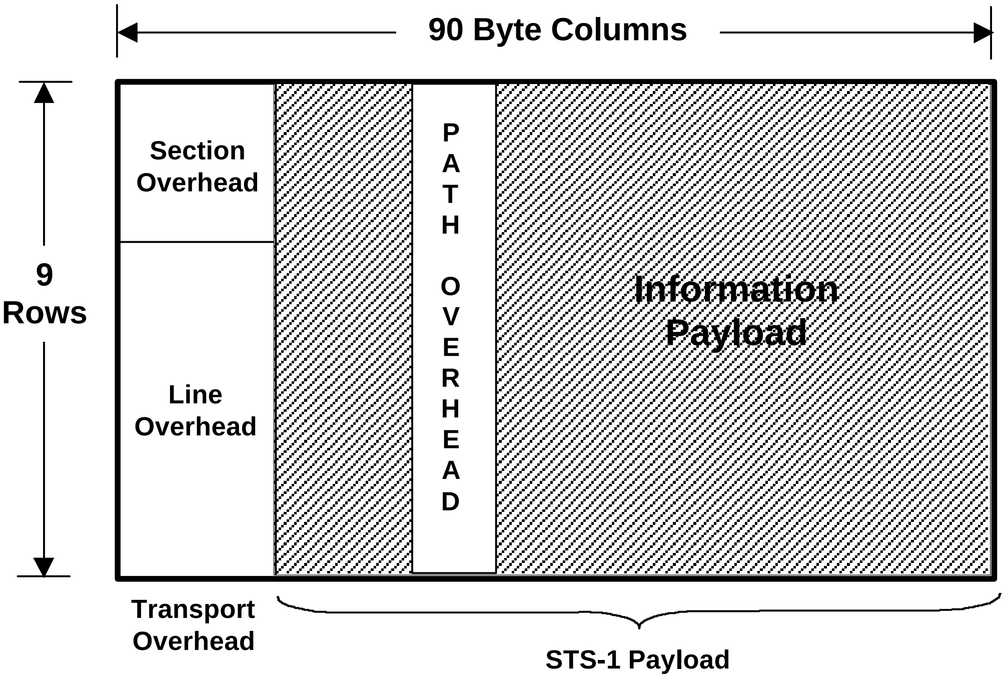


Figure 3.2 SONET STS-1 Frame Structure

The transmission of the actual frames occurs from the upper left to the lower right, just as if reading text on a page. It can be seen from Figure 3.2 that the first 3 byte columns of the SONET frame are dedicated to the Transport Overhead (TOH), which closely corresponds to the Section Overhead (SOH) – the first 9 byte columns in SDH STM-1 frame. Both TOH and SOH have a very important role in supporting transport capabilities such as framing, error monitoring, and management operations.

In SONET, the overhead and transport functions are grouped into layers. SONET layers, section, line, and path, have a hierarchical relationship, and can be viewed either from the top down or from the bottom up. (Figure 3.3) Each layer has its own management communication through the specific octets in the overhead.

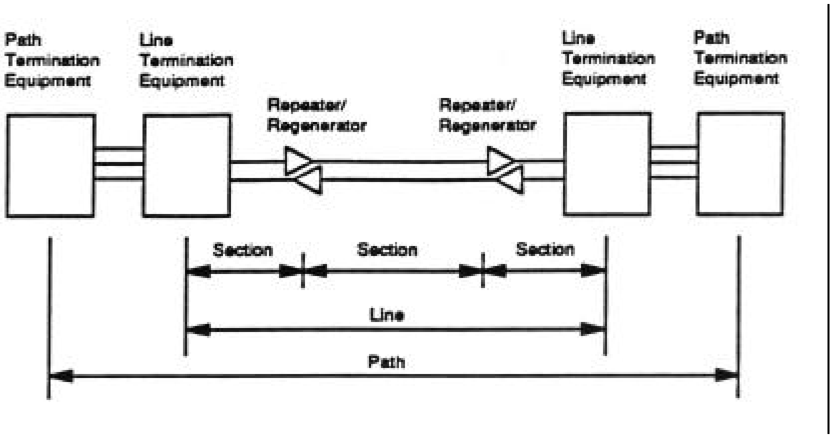

Path Termination Equipment
Line Termination Equipment
Repeater/ Regenerator
Repeater/ Regenerator
Line Termination Equipment
Path Termination Equipment
Section
Section
Section
Line
Path

Figure 3.3 Paths, Lines and Sections

The TOH, which is composed of Line overhead and Section overhead, is added to the Synchronous Payload Envelope (SPE) to create the STS-1 frame. The SPE is further composed of the POH and the information payload. The functions for TOH and POH are given in Table 3.2.

Table 3.2 TOH and POH Functions

Transport Overhead Functions	**Path Overhead Functions**
Framing	Path trace
Error detection	Error detection
Orderwire	Payload composition
User channel for network provider data communication (for operations)	Maintenance signaling
Pointer (for locating payload)	Far-end error information
Automatic protection switching signaling	Path user
Maintenance signaling	Payload-specific “signaling”

3.2.1. SONET Multiplexing and Mapping

Although the STS-1 signal is the smallest building block in SONET, most of the end user devices do not operate at the rate of 51.84 Mbps. If the path terminating equipment does not need the entire SPE, the multiplexer can define smaller/slower paths within the SPE called Virtual Tributaries (VT) in SONET or Virtual Containers (VC) in SDH. The multiplexing of slower rates into SONET STS-1 is different and comparatively simpler than the multiplexing of slower rates into SDH STM-1 signals. It should be noted that SDH uses different standards for the slower signals, and is somewhat more complex than SONET in that it allows different mappings of the same payload, but also requires four hierarchical levels in forming the payload of an STM-1.

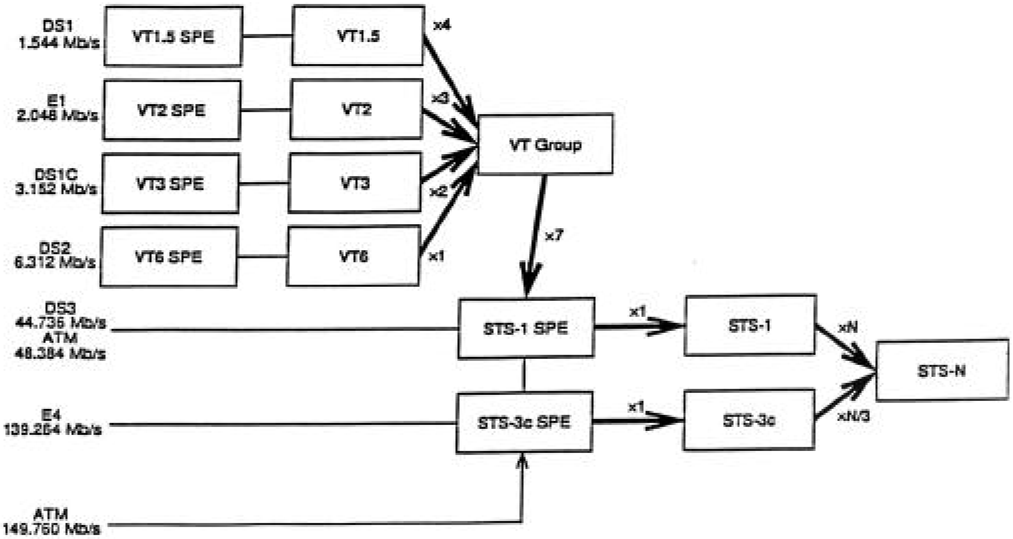


Figure 3.4 SONET Multiplexing Structure

The first step in SONET multiplexing is to map the slower signals into a Virtual Tributary Synchronous Payload Envelope (VTx-SPE) which corresponds to the appropriate size VT. Achieving greater SONET/SDH rates than the basic building blocks is more straightforward than mapping slower signals into the STS-1/STM-1 signals (See Figure 3.4)[6]. For SONET, higher rates are accomplished by byte-interleaving N frame-aligned STS-1 signals to create an STS-N signal. Since there is no overhead added during this multiplexing process, the rate of a newly created STS-N signal is exactly Nx51.84 Mbps, where values of N in current use are 1, 3, 12, 24 and 48.

However, when mapping high rate signals directly onto an STS higher than STS-1, such as ATM, then the frame aligning must be done with pointer concatenation which results in an STS-Nc concatenated frame signal with a locked (concatenated) STS-N payload (See Figure 3.5) [3]. After mapping these signals into a big enough STS-Nc SPE, they could be further multiplexed into an STS-N signal.

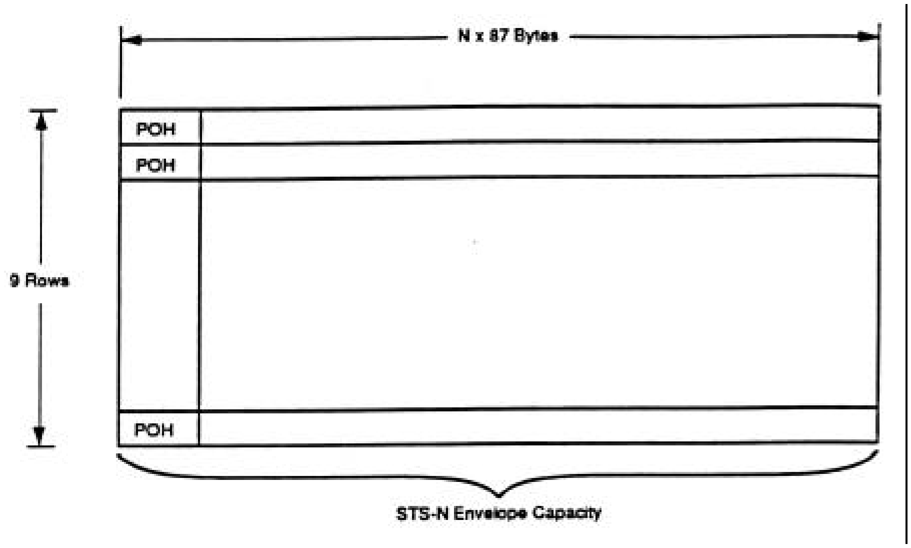


Figure 3.5 Concatenated STS-Nc Payload

3.3. Asynchronous Transfer Mode

The ATM concept can be approached from many different aspects but in its most essential sense ATM is a technology which is defined by standardized protocols. ATM is a connection oriented, cell-based switching and multiplexing technology designed to offer flexible transmission support for a wide range of services, including voice, video and data. ATM is connection oriented because it communicates with its peers by establishing virtual connections in a similar fashion to typical circuit-switched networks. It is a cell-based technology because ATM partitions all of its information into fixed-size packets or cells, which in turn simplify the switching and multiplexing functions. In addition to supporting a complete range of user traffic, simplified switching and multiplexing capabilities, ATM is well suited for high-speed networking too. This makes it the underlying technology of choice for B-ISDN.

3.3.1. B-ISDN PRM

The ITU-T's B-ISDN protocol reference model, including ATM as its foundation, is shown in Figure 3.6 [9]. The B-ISDN protocol reference model (PRM) may be a useful model for conceptual visualization of certain functions which define ATM, but it certainly isn't the only one, nor do the manufacturers stick strictly to it when implementing ATM switches and interfaces. One of the reasons for this is that there are other standards setting bodies that define and further shape ATM. In addition to ITU-T, ANSI, and European Telecommunications Standards Institute (ETSI), one body that emerged as the dominant player in the ATM arena is the ATM Forum, founded by Northern Telecom, SPRINT, SUN Microsystems, and Digital Equipment Corporation (DEC) in 1991 [5].

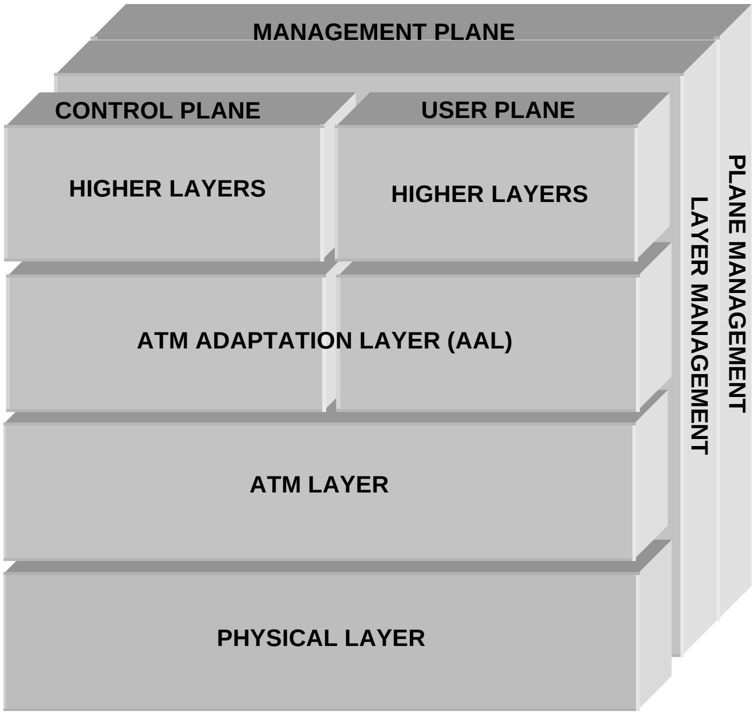


Figure 3.6 B-ISDN Protocol Reference Model

3.3.2. ATM Planes

The B-ISDN model consists of the three planes -- user, control and management planes -- which are labeled on top and stretch over the front and sides of the cube pictured in Figure 3.6. The user plane is responsible for transferring user information from applications, and the control plane is responsible for call and connection functions necessary for establishing the switched services. They both accomplish their functions by making use of the common underlying ATM and physical layers, as well as utilizing some of the same AAL layer protocols. The management plane is responsible for providing capabilities for exchanging information between the control and user planes, and is further divided into the layer management and plane management. The layer management is specifically responsible for management of functions specific to each layer, whereas plane management manages the entire system as a whole.

3.3.3. ATM Layers

Following its layered modeling practice, ITU-T offers further decomposition of the B-ISDN PRM into layers, and the detailed descriptions are contained in ITU-T recommendations I.321 and I.413 [9].

3.3.3.1. Physical Layer

The physical layer provides for transmission of ATM cells over a physical medium connecting ATM devices; it is divided into two sublayers, the Physical Medium Dependent (PMD) and Transmission Convergence (TC) sublayer, as shown in Figure 3.7. One of the reasons for sublayers is to decouple the transmission from the physical medium to allow for many different physical media [13].

The PMD sublayer includes bit generation, transmission and reception capabilities and bit alignment. Line coding and possible electro-optical conversion are also functions of the PMD, as well as all other medium specific functions required for transmission and reception of data such as voltage levels, frequency, light intensity, wavelength, etc. For this project, the physical medium is multimode fiber. All timing, synchronization and framing methods are done according to SONET specifications.

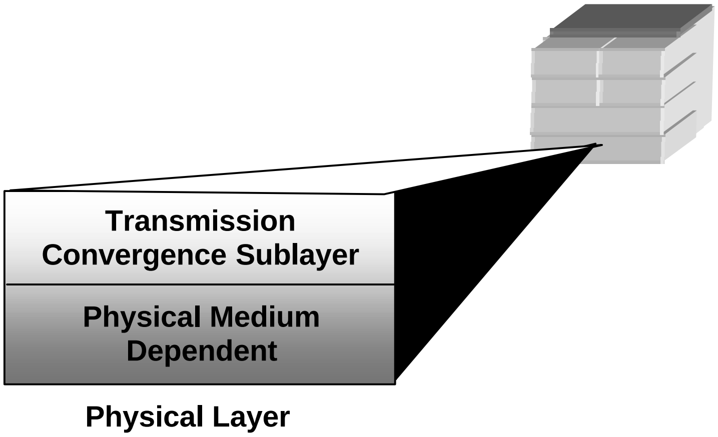


Figure 3.7 Physical Layer of the B-ISDN PRM

The TC sublayer maps the ATM cells to and from the PMD sublayer. In this case TC is responsible for mapping the ATM cells into SONET's OC-3c SPE. Since the integer number of ATM cells does not fit into the SPE, the TC sublayer continuously maps the cells across the SPE's boundaries. However, on the receiving end the TC needs to perform cell delineation, which is a process of recovering the ATM cells boundaries out of the continuous bit stream now arriving from the PMD.

Two functions helpful for cell delineation are scrambling / descrambling of the cell's information field and generation / verification of the Header Error Check (HEC). Scrambling / descrambling prevents the cell delineation mechanism from malicious attacks of bundled errored blocks of bits. Generation / verification of the HEC is a one-byte code applied to the header, capable of correcting any single-bit error in the header as well as detecting many patterns of multiple bit errors. In addition to these, the TC sublayer is also responsible for cell-rate decoupling, which is insertion at the transmission side and then suppression on the receiving side of the idle cells, which helps maintain a continuous flow of cells for the PMD-specific rate [5, 9, 13].

3.3.3.2. ATM Layer

The ATM layer can be further divided into the Virtual Channel (VC) level and Virtual Path (VP) level as shown in Figure 3.8.

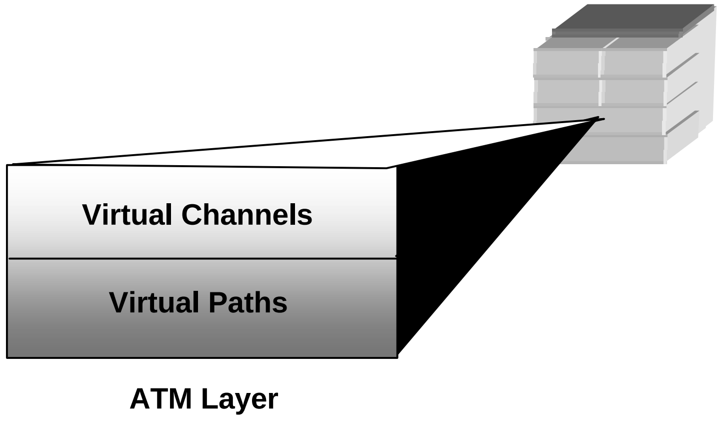


Figure 3.8 ATM Layer of the B-ISDN PRM

The VPs and VCs are instrumental to the ATM operations, and are defined in ITU-T Recommendation I.113 as [9]:

VC: "A concept used to describe unidirectional transport of ATM cells associated by a common unique identifier value."

VP: "A concept used to describe unidirectional transport of cells belonging to virtual channels that are associated by a common identifier value."

The relationship between VPs and VCs is shown in Figure 3.9.

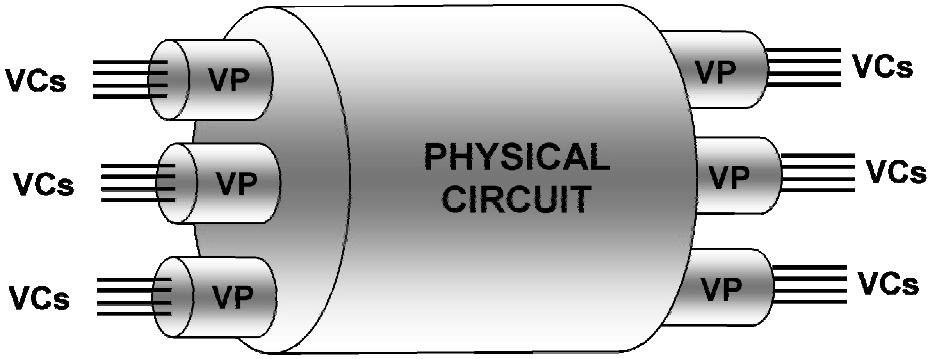


Figure 3.9 ATM Physical Circuit, VPs and VCs

It is the ATM layer that contains the functions that are most unique to ATM technology. First, at the transmitter, cells from individual VPs and VCs are multiplexed into one composite stream. At the receiving side, cells arriving from a composite stream are demultiplexed into individual cell flows designated for individual VP and VC cell streams.

Second, the ATM layer at the switching/routing nodes performs Virtual Channel Identifier (VCI) and Virtual Path Identifier (VPI) translation. VPIs and VCIs are the unique identifier values mentioned in the above ITU-T's definitions for VPs and VCs. VPI identifies a bundle of one or more VCs, and VCI identifies one unique VC in a particular VP. VCIs and VPIs are included in the ATM cell headers and are used for establishing end-to-end connections between ATM end devices as well as for necessary routing and switching during the connection setup phase.

VPIs and VCIs have only a local significance and therefore can be reused [5]. The ATM layer of each ATM device in a network between the end users assigns VPIs and VCIs necessary for creating an end-to-end VP or VC. Since it is possible that each switching node already uses a certain VPI or VCI, it is this case where the VPI/VCI translation is needed. In other words, switches are allowed to translate the values of individual VPIs and VCIs to any available value as long as that value describes the unique VP or VC and preserves the integrity of the end-to-end connection.

The third function of the ATM layer is generation/extraction of the header. After receiving cell information field from the higher layers, the ATM layer generates and adds an appropriate header, except for the HEC value, which is done by the physical layer. On the receiving side, it removes the header and passes the cell information field to the appropriate higher layer.

Finally, the ATM end device connecting to the network through the User to Network Interface (UNI) utilizes a Generic Flow Control (GFC) parameter which helps with the control of the ATM traffic flow in a customer network [9].

3.3.3.3. ATM Adaptation Layer

The AAL layer offers the services provided by the ATM layer to the higher layers. Most of the higher layers hand down their information in variable size packets, or have some other unique bit rates or information transfer requirements. However, no other service communicates in 53 byte cells so it is the AAL's function to adapt all these different services to fit the ATM's 48 Byte information field or payload. In order to accomplish its task, the AAL's functions are divided into the Segmentation and Reassembly (SAR) Sublayer and the Convergence Sublayer (CS), as shown in Figure 3.10.

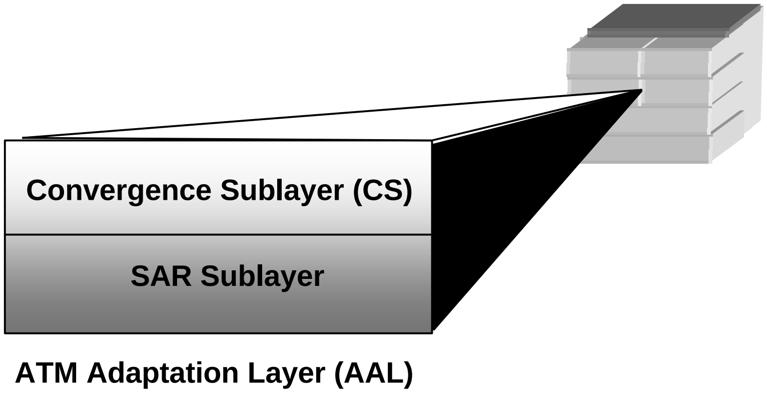


Figure 3.10 ATM Adaptation Layers of the B-ISDN PRM.

It is the SAR sublayer that is responsible for segmentation of higher layer information into the format suitable for consecutive ATM cells for a certain VC. The SAR layer is also responsible for extracting the higher layer information from the incoming ATM cells from a particular VC and reassembling them into a format usable by the higher layers.

The CS layer is responsible for identifying individual messages, recovery of timing, flow control, etc. It is further subdivided into the Service-Specific (SS) and Common Part (CP) sublayers. The SS sublayer performs certain functions particular to higher layer AAL user or, in absence of need for such functions, it may be null [5]. The CP layer, however, must always be complemented with the lower SAR layer, and is responsible for passing off SAR Protocol Data Units (PDU), which are essentially the ATM cell payload, to and from the ATM layer.

In general it is the AAL which needs to interface with an ever-expanding set of higher layer functions. The AALs were geared toward different classes of service, as shown in Table 3.3, and were classified based on the time relationship between source and destination, bit rate and connection mode [14].

Table 3.3 AAL Service Classification [15]

	CLASS A	**CLASS B**	**CLASS C**	**CLASS D**
Timing Relation between Source and Destination	Required		Not Required	
Bit Rate	Constant	Variable		
Connection Mode	Connection Oriented			Connectionless
AAL Protocol	Type 1	Type 2	Type 3/4 Type 5	Type 3/4

So far there are five AAL protocols than can provide different classes of service to the higher layers [9].

1. **AAL-0** is an AAL with empty SAR and CS, therefore adds no functionality between the ATM layer and higher layers, allowing information to be transparently transferred through the AAL.

2. **AAL-1** is generally recommended for constant bit rate applications sensitive to cell loss, delay and jitter, such as voice and constant rate video. It is also very appropriate for emulating constant-rate leased lines.

3. **AAL-2** provides transfer of variable bit rate applications with preservation of timing relationship between source and destination. It is commonly intended for the transport of variable bit rate compressed voice and video.

4. **AAL-3/4** was created by merging what were two separate AALs intended for two separate classes of service. However it was possible to merge the two AALs into one, which is able to satisfy demands from both classes of service. AAL-3/4 is mainly intended for applications that are sensitive to loss but not to delay. It can be used for both connection oriented as well as connectionless services.

5. **AAL-5** is mainly intended for data traffic although it has recently been used for all classes of service. AAL 5 offers same services as the AAL-3/4, but it does so with less overhead, no support for multiplexing functions and less complexity.

3.3.4. ATM Resources

The combination of fixed cell sizes, concept of virtual paths and virtual channels and five different categories of ATM adaptation layer protocols makes ATM one of the most flexible high speed networking technologies implemented today. In addition to it's flexibility for supporting voice, video and data and ability to scale to large networks, ATM is the first and so far the only widely implemented technology that can offer Quality of Service (QoS). Some of the ATM resources responsible for these achievements are discussed below.

3.3.4.1. ATM Cell

The transmission, switching and multiplexing unit of ATM technology is the 53-byte, fixed length packet, or cell. There are two formats for ATM cells, and which format is used depends on where they are used in the network. The traffic between the end user and the ATM network is carried on UNI cell format while the internetworking is done over the Network-to-Network Interface (NNI) cell formats. Both cells however are identical size of 53-bytes and both consist of 48-bytes payload and 5-bytes header, as presented in Figure 3.11 [5].

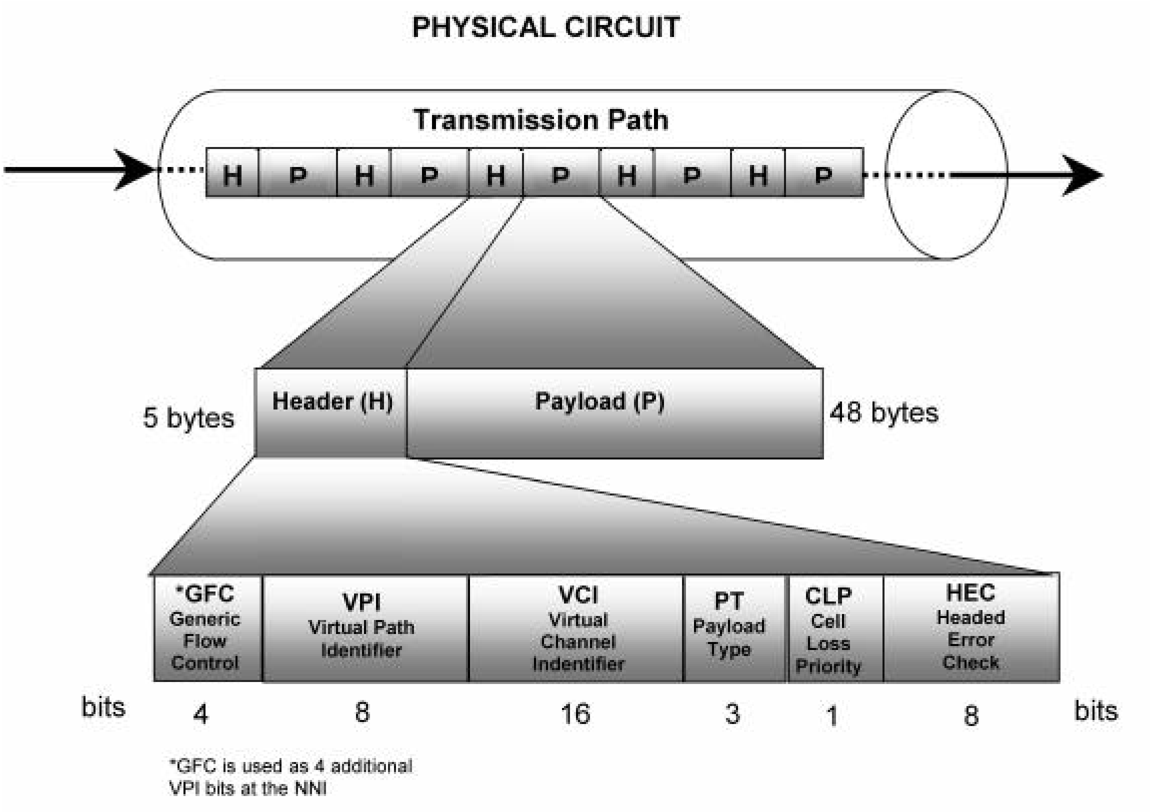


Figure 3.11 ATM Cell Format

The only difference between the two formats is the four-bit GFC parameter contained in UNI cell format, which are replaced by additional VPI field in a NNI cell format.

There are somewhat different stories behind the reasons for the 48-bytes payload size but all of them agree that it was an eventual compromise between the 32-Bytes and 64-Bytes sizes. The main reason for debate, however, was focused on the basic tradeoff between the smaller 32-bytes cells and larger 64-bytes cells. The smaller 32-bytes cells were favored in Europe where the small country sizes would enable the European telephone companies to achieve low enough latency in packing and transporting the voice via cell without the need to install echo cancellation devices. On the other hand the size of the US forces the phone companies to install echo cancellers anyway, so the large 64-bytes packets would be preferable since they would offer higher overall transmission efficiency by improving the header-to-payload ratio [6]. There is a mention that the size of the header was a separate tradeoff on its own between the increased function through an 8-bytes header and increased efficiency through a 3-bytes header [5].

As mentioned earlier, the switching and multiplexing of a cell is done by using VPIs and VCIs ,which are contained at the beginning of the cell header and allow for finding routes from input ports to output ports through individual switches by VPI/VCI translation, as well as establishing the end-to-end connection.

The next header field is the three-bit Payload Type (PT) Indicator which basically differentiates between the control data or the user data being carried in the payload of the cell. Aside from user data, the PT indicates VC Channel (VCC) -level Operation and Maintenance (OAM) information, Explicit Forward

Congestion Indication (EFCI), AAL information and the Resource Management (RM) information [16].

The Cell Loss Priority (CLP) bit differentiates between the high and low priority traffic and it is a topmost layer of priority setting for the traffic that has already passed through other congestion management systems [11]. For example, this bit can be set for low priority either by an application which could afford to lose certain cells or by a network element which encounters cells transmitted in excess of what was initially negotiated [6].

Finally the HEC, which uses an eight -bit Cyclic Redundancy Checksum (CRC) polynomial performs an error detection and single error correction on the header.

3.3.4.2. Quality of Service

As is described in section 3.3.3.3, ATM provides 5 classes of service, which in turn can offer support to most existing data services and their various traffic patterns. Certain services can tolerate some cell loss while others cannot. Other services have timing constraints while some do not. And some may have multiple constraints while some have none. Thus, in order for ATM to be able to support all the different services, while using available network resources efficiently and providing specified and guaranteed levels of Quality of Service (QoS), it needs to bound and control the different traffic parameters associated with each service.

3.3.4.3. Contract Parameters

During the connection set-up a source can specify the following traffic parameters [17]:

Peak Cell Rate (PCR) represents the maximum instantaneous rate at which the user will transmit. This parameter is the inverse of the minimum cell inter-arrival time.

Sustained Cell Rate (SCR) is the average rate for a given connection over a long time period.

Cell Loss Ratio (CLR) is the ratio of the number of lost cells sent by a source to the total number of cells sent by the source, as presented in Equation 3.1.

$$\text{CLR} = \frac{\#\ \text{Lost Cells}}{\#\ \text{Transmitted Cells}}$$

Equation 3.1

The cells are usually lost in the network due to error or congestion and buffer overflows. If congestion is the reason then the network will first discard the low priority cell or cells with CLP=1. Nevertheless, CLR can be set separately for both the low priority cells and high-priority cells.

Cell Transfer Delay (CTD) is the total delay experienced by the cell from the time it enters the network until it leaves the network. This parameter includes propagation delay, queuing delays at intermediate ATM devices and service times at queuing points.

Cell Delay Variation (CDV) is a measure of variance of CTD. It is a variation of delay or jitter experienced by a cell propagating through the network.

Burst Tolerance (BT) is the maximum burst size that can be sent at the PCR and it is used to control the traffic entering the network via the leaky bucket algorithm which puts all arriving cells in a buffer (bucket) and sends them at the SCR. The Maximum Burst Size (MBS) is the maximum number of back-to-back cells that can be sent out of the bucket at the PCR. The relationship between the BT and MBS is shown in Equation 3.2.

Minimal Cell Rate (MCR) is the minimum rate that the user desires.

$$BT = (MBS - 1) \left(\frac{1}{SCR} - \frac{1}{PCR} \right) \qquad \text{Equation 3.2}$$

3.3.4.4. Service Categories

Once the traffic parameters of a service are made known to the ATM network, but before the connection through the network is established, a contract for desired QoS based on the traffic parameters presented is being negotiated. If all intermediate ATM devices en route from source to destination can support the requested QoS, the connection gets established and all the traffic requirements for that particular service are guaranteed for the duration of the connection. In a broad sense all traffic contracts specify one of the five service categories defined by the ATM Forum so far. The five service categories are [5]:

Constant Bit Rate (CBR) service category supports real-time applications, which require a fixed amount of bandwidth defined by the PCR. CBR is best suited for applications with stringent requirements on CTD and CDV such as voice and the constant-bit-rate video. When the CBR request is made to the network, all switches and intermediate devices must be able to provide the specified PCR or the connection will be refused.

Real-Time Variable Bit Rate (rt-VBR) service category allows for rate variation while preserving high demands on CTD and CDV. The rate variation is bounded by the PCR and the "average" rate defined by the SCR and MBS. The three parameters, PCR, SCR and MBS, define a worst case scenario for which the given QoS traffic contract will hold. The applications usually requesting rt-VBR are delay-variation-sensitive and possibly bursty in nature. Variable-bit-rate video is an example of such traffic.

Non-Real-Time VBR (nrt-VBR) is a service category geared toward applications bursty in nature but with no constraints on delay and jitter. The traffic contract is the same as the one for the rt-VBR and is specified by PCR, SCR and MBS. Applications such as airline reservations, banking transactions and other terminal sessions would use nrt-VBR.

Unspecified Bit Rate (UBR) is also known as the "best effort" service, since it does not specify any bandwidth, has no guarantees for throughput, places no constraint on delay nor jitter and therefore offers no QoS. Since most LANs and IP networks also offer only best effort service, LAN emulation, IP over ATM and other non-mission-critical applications use UBR. In general UBR is a good service category for handling applications which have built-in retransmission schemes [18].

Available Bit Rate (ABR) service category uses a closed-loop flow control, which together with the sources that vary their transmission rates based on the network feedback, tries to dynamically vary the traffic flow in order to use all the available bandwidth in the network not used by the other service categories, as shown in Figure 3.12 [18].

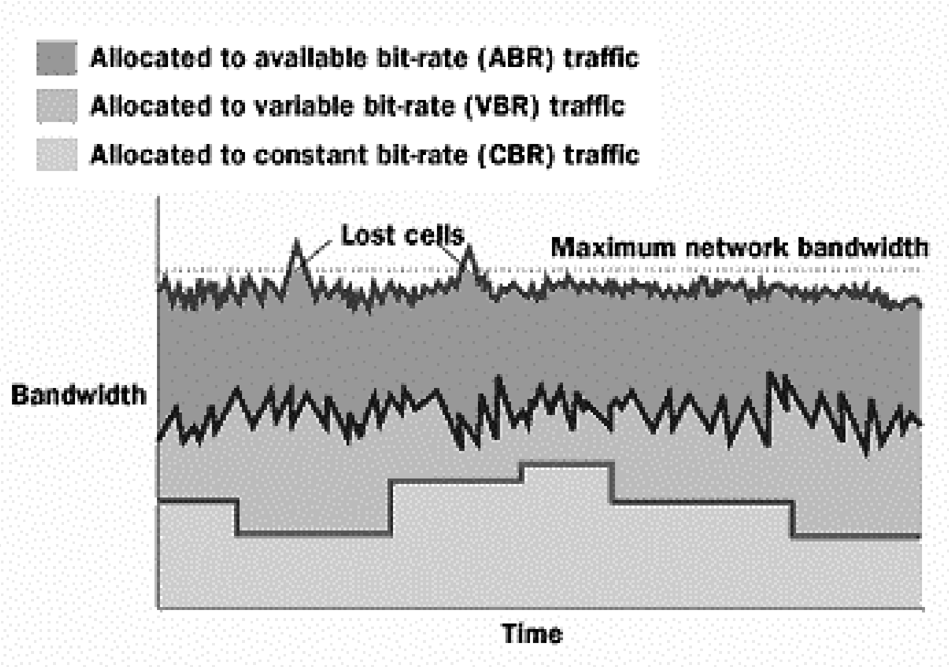


Figure 3.12 Bandwidth Allocation for Different Service Categories [18]

Thus, for users that can change their transmission rates, ABR can offer a relatively low loss service. The ABR service contract is based on the PCR and MCR parameters but offers no delay and jitter constraints. Therefore non-real-time applications such as LAN interconnections, web browsing, database archival and file transfers are good candidates for ABR.

3.3.4.5. Congestion Avoidance and Flow Control

As indicated previously, ABR and UBR support similar services but due to rate-based flow control protocols for avoiding congestion specified for ABR, ABR is becoming a preferred way for sending data traffic over ATM.

So far there are four options for rate-based flow control and they all rely on specialized resource management (RM) cells which provide feedback and some information about the congestion state in the network [19].

Explicit Forward Congestion Indication (EFCI) is the simplest scheme for congestion avoidance. Any ATM node that is experiencing congestion can set the EFCI to indicate to the upstream nodes, and ultimately to the end-user, that once the EFCI is set no other network element can alter its value. The node sets EFCI to indicate congestion by setting first two bits of the PT field to 01, and once the end-user at the destination receives this indication it is up to higher layers to act, for instance by lowering the transmission rate at the source [15].

Relative Rate (RR) marking is a rate-based method for flow control that uses Resource Management (RM) cells. Any node can set the Congestion Indication (CI) bit or the No-Increase (NI) bit in either forward or reverse direction of the traffic flow. As in the EFCI case, once any switch or node sets either one of the CI or NI bits, no other network element can alter them. When the source receives the CI bit, it decreases its rate, whereas the NI bit prevents the source from increasing the rate.

Explicit Rate (ER) marking is another end-to-end flow control scheme and it uses the ER field in the RM cells for conveying information from the network nodes back to the source. The source sets the ER field to reflect the rate agreed on with the network at the connection set-up negotiation process. Usually the end-user requests the PCR; but since this is ABR, and there are no quality guarantees, the switches can change the ER in RM cells flowing in either direction to better reflect the resources available to that particular connection. When the source receives the RM cell with altered ER field, it has to adapt its transmission rate to the rate specified by the ER.

Virtual Source/Virtual Destination control uses the same principles as listed above, except that instead of being an end-to-end flow control scheme, the control loop is broken into two or more segments, and the intermediate ATM nodes act as virtual sources and virtual destinations. In this way each segment can be treated as a smaller end-to-end problem and the congestion problem is, or can be, controlled on every hop.

If flow control fails, however, congestion can occur, and ATM has few congestion recovery techniques that may aid in restoration of normal operating conditions [5].

Selective Cell Discard is a mechanism that allows the network to discard cells with low priority (CLP=1). This method can be a powerful tool for aiding congestion recovery, especially when used with Usage Parameter Control (UPC) tagging. UCP would change the CLP bit to indicate low priority for all cells that do not conform to the traffic parameters negotiated at the connection set-up.

Early/Partial Packet Discard (EPD/PPD) is a scheme that maximizes the transfer of good packets by discarding on the packet level instead of the cell level. The main reason is that cells are generally much smaller than the higher layer variable-size packets and it takes many cells for transferring an entire packet. However, if one cell gets discarded then the SAR cannot complete the reassembly of the packet, and the request is usually sent for a retransmission of the entire packet -- not just the cell missing. Thus it is much better to free the buffer of the whole group of cells comprising the packet.

EPD prevents all the cells from a particular packet that has not entered the buffer yet from entering the buffer, and therefore reserving the buffer space for cells belonging to packets already admitted to the buffer. If a cell belonging to a certain packet already in the buffer gets dropped due to congestion, then PPD discards all remaining cells belonging to that packet, except the last one.

Disconnection is another, and rather drastic, approach which can be used in times of prolonged severe congestion. In this scheme certain connections would be dropped regardless of the contracts. An example is the US government, which usually places requirements upon carriers to drop all other traffic if necessary in order to provide resources for priority national defense traffic or local emergency services.

3.4. LAN Emulation

There are obviously numerous advantages offered by ATM. However, most data traffic in existing Customer Premises Networks (CPN) is carried over LANs, Ethernet (IEEE 802.3) and Token Ring (IEEE 802.5) in [12]. In order for all disparate internet devices to operate over ATM, LANE is used to make ATM appear as if it were a traditional Ethernet or Token Ring. Basically, LANE disguises an ATM network so that it appears as a traditional network to the users and applications.

Disguising an ATM network to resemble a Legacy LAN is not an easy task considering that traditional LANs operate quite differently than ATM. Ethernet and Token Ring are both connectionless unlike the ATM which is connection oriented. The shared medium of LANs, in turn, make multicast and broadcast easily accomplished which isn't the case with ATM. Furthermore, Media Access Control (MAC) addresses used in LANs are burned in by the manufacturer and are completely independent from the network topology [12].

In order to overcome these major differences LANE is comprised of three major components [17]:

LAN Emulation Client (LEC) maps the MAC address into an ATM address and then sets up an ATM channel between that LAN device’s ATM address and an ATM address that corresponds to the MAC address of the target LAN device. The ATM channel is established by creating a direct VC between the corresponding ATM addresses followed by the necessary assignment of the VCIs.

LAN Emulation Server (LES) is a LANE software entity that is mainly responsible for support of the LAN Emulation Address Resolution Protocol (LE-ARP). If a source LEC does not have the destination address in its address resolution table, it sends the LE-ARP request to the LES for the ATM address of the target LEC responsible for a certain destination MAC address.

Broadcast / Unknown Server (BUS) has a main task of forwarding messages to certain LECs or broadcasting messages to all LECs attached to the given network.

3.5. Ethernet

In this project, Ethernet (IEEE 802.3) was emulated to run over the experimental laboratory ATM network. Ethernet is a very common LAN technology, which originated from early packet-radio networks, that uses Carrier Sense Multiple Access/ Collision Detection (CSMA/CD) over a 10 Mbps shared link. Basically Ethernet communicates across a medium shared by many users and thus it is a multiple access technology. "Carrier Sense" means that all users are capable of distinguishing between a busy or an idle link, and they can "listen" to determine whether they can transmit or not. "Collision Detection" means that the nodes can also "hear" whether the frame they just sent collided with some other user's frame. If so, the node will try to retransmit the frame after some random waiting period [6].

Generally, host adapters pack the data into a variable size payload of the Ethernet frame, which includes a CRC at the end of the frame and attaches the header at the beginning of the frame. The header is used to synchronize with the signal on the shared link, to include the destination and source address and to let the destination know which higher protocol the data is intended for and what type of data is contained in the payload, the maximum size of which is 1500 bytes.

3.6. Internet Protocol

Internet Protocol (IP) is a protocol that provides connectionless, best effort service of IP packets or datagrams across the Internet. Its wide acceptance is a result of its simplicity and ability to run over any kind of network.

However, this simplicity has a drawback: IP offers an unreliable service since IP just sends the packet with an included source and destination address, and then lets the network attempt to deliver it. Using the destination address from the header, the intermediate switching and routing devices know where to forward the packet next. The IP packets are also variable size with a maximum of 65535 bytes. In order to preserve flexibility and run over any network, IP supports fragmentation and reassembly so that its datagram size can always fit within each underlying network's Maximum Transmission Unit (MTU).

If the packet gets lost, however, IP has no way of telling, nor does it know, if there is congestion or even if the destination exists. If the packets don't get lost, they can still be delivered out of order, or sometimes any one packet can be delivered more than once. All in all, IP leaves it up to the higher layer protocols to handle these failure modes [6].

3.7. User Datagram Protocol

User Datagram Protocol (UDP) is an end-to-end transport protocol of the Internet that provides connectionless datagram services for processes-to-process communications between two or more hosts.

The main function of such a protocol is to be able to differentiate between multiple application processes running on a particular host, and be able to connect the same processes running on different hosts separated by some form of network. To successfully perform this function, UDP includes an identifier port on which the particular process is running for both the sender and the receiver of the message. Aside from this demultiplexing function, UDP offers only a "best effort" service, and implements neither flow control nor reliable/ordered delivery [6].

3.8. Transmission Control Protocol

Transmission Control Protocol (TCP) is, like UDP, a transport protocol. TCP also must allow multiple application programs on any given host to communicate simultaneously with its peers by supporting a demultiplexing mechanism. However, unlike UDP, TCP is a connection oriented, reliable/ordered delivery service, which implements a very efficient congestion control mechanism.

This mechanism in particular puts a limit on how fast the applications running over TCP can send data. Since TCP is a full-duplex protocol, supporting a connection in each direction across the network, it has feedback capabilities which can determine if there is a congestion problem in a network, and relay this data to the sender in order to slow down the transmission rate if needed [6]. In addition these functions can be performed and parameters can be changed dynamically for the duration of a particular connection, making the TCP very immune to constantly changing network conditions. This immunity was also manifested in this project.

3.9 The Big Picture

The setup assembled for this particular research project consists of most of these protocols stacked on top of each other as shown in Figure 3.13.

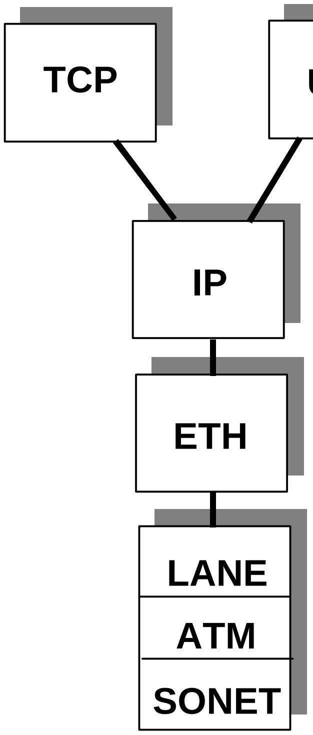


Figure 3.13 Protocol Stack

The application scripts described in Chapter 5 deliver data which represents a protocol data unit (PDU) to either UDP or TCP, which performs a variety of operations and adds its own header to create a segment. This segment is then passed down to IP which also performs certain operations and adds its own header creating a datagram in Internet terms [20]. This datagram is next passed down to Ethernet which appends its own header and trailer thereby forming a packet. LANE then takes the Ethernet packets, converts its MAC address into an ATM address, performs segmentation of the payload contained in the Ethernet packet, and adds an ATM header and trailer, thus forming ATM cells. The ATM cells are then placed within SONET's SPE which adds the POH, and finally forms a SONET frame. SONET frames are then carried across the network to the other host where processes are implemented in reverse.

It should be noted that partial decapsulation occurs in intermediary nodes, the two ATM switches in this particular case, which are needed for reading the addresses and further routing through the network toward the final destination.

This chapter points out that some of these protocols are connection-oriented and some are connectionless. Some have error checks over their headers while others have them over headers and payloads. Some are designed for simplicity and multiple access while others are complex and have a multitude of functions to provide reliable service.

All in all, it is a complex mix, which may have only one thing in common: the fiber. It is the fiber that ends up ultimately carrying all the information and is assumed to always perform well; if there is a problem, then it is assumed to be in higher network layers. In part because of this and the complexity of today’s computer networks, it was decided in this project to investigate that assumption.

The goal is to gain some insight into whether or not user-level performance varies under slight degradations in the physical medium, and further how the degradation affects performance. The results could offer a better understanding of the interactions between network protocols and network elements. Perhaps the results will also yield that certain methods of configuring network parameters are better than others, and finally, what, if anything, should be done to avoid certain conditions or improve certain situations.

Chapter 4. Network Failures

Network performance has become a critical issue in the area of computer networks and the contemporary communications field. There is a greater need than ever before for reliable service due to several factors: an increasing dependence of end-users on telecommunications; a greater demand for high-quality performance; the need for assured service; and the special reliability needs for government, emergency and military organizations. These demands have resulted in much work on categorizing, classifying, and analyzing network failures as well as studies in areas of network survivability performance, network survivability characterization and restoration [20].

4.1. Classification

Although somewhat ambiguous, classification of failures has received a lot of attention from the research community and telecommunications standards-setting bodies. Based on the work of ANSI Technical Subcommittee T1A1, three features of an outage in a network can be used to describe outage categories [1].

The three features are: 1) unservability (U), which represents a percentage of failed links or circuits in circuit switched networks, percentage of lost or unusable packets delivered in a packet switched network and a percentage of failed service units for leased lines; 2) duration (D), or a time period for which unservability lasts and 3) weight (W) which represents a population or geographic area in which the unservability exceeds a certain threshold [1].

The (U, D, W) triple can be graphed in three-dimensional space. The values of U, D and W for a particular failure would then define a region as shown in Figure 4.1 [1].

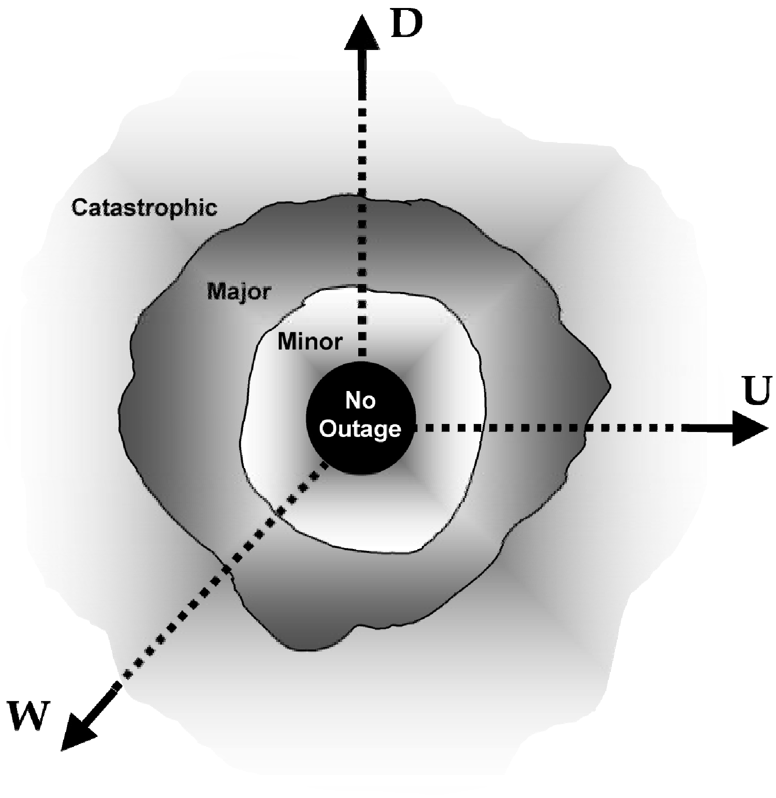


Figure 4.1 Failure regions for U, D, and E

The failure categories that can be defined based on the three features are catastrophic, major or minor [1]. Catastrophic network failures are often triggered by such disastrous occurrences as earthquakes, floods, tornadoes, power grid failures, or war. Major cable backbone failures can happen as a result of fire, terrorism, or tandem switch failure. Minor single link or path failures often are due to fiber failure, equipment failure, and shelf or unit failure. It should be pointed out that these failures are all hard failures and assume a complete failure of a part of a network no matter how small a role the part plays.

4.2. Restoration Techniques

Today, most of the solutions for these hard failures have been incorporated in the lower layers of networks, particularly in SONET/SDH. ITU-T has released many recommendations for SONET/SDH restoration techniques such as automatic protection switch (APS), self-healing ring (SHR) and self-healing network (SHN). ITU-T has also begun working on standardizing restoration and self-healing techniques for ATM networks, which include slight modifications of the same methods implemented in SONET/SDH, as well as new ones unique to ATM, such as failure-resistant virtual path (FRVP) [21].

The main drive behind ITU-T's continued work toward expending restoration capabilities into the higher network layers is the one way dependency between the protocol layers. In other words, the lower layers cannot see the defects occurring at the higher ones while the higher ones will always be affected by defects in the lower layers. Thus, SONET/SDH offers very fast restoration performance from fiber failures but does not protect against defects in the ATM layer because ATM cell defects are invisible to the SONET/SDH defect-detection mechanism [22]. As newer technologies emerge as potential new layers there will be a continuing need to study and later standardize their survivability/restoration schemes, as well as define their roles in the multi-layered networks as a whole.

4.2.1. Single-Layer Network Restoration

Just as classification of network failures is somewhat ambiguous, so is the classification of network restoration methods. The two largest categories that cover most, if not all network restoration methods are centralized and distributed control schemes.

4.2.1.1. Centralized Control Schemes

Centralized control schemes include one main center that oversees the entire network and is responsible for the entire network survivability process, from detection of failures to finding an alternate route. Since the control center has an overview of the entire network, it has a better chance of finding the best alternate route, as well as distributing the extra load from the failed network elements most efficiently. Centralized control schemes can be implemented in SONET/SDH as well as in an ATM layer. However, the big drawback is the complexity of such centralized management, and the relatively slow speed of restoration. Examples of implemented centralized control schemes are AT&T's FASTAR and NTT's SUCCESS [23, 21].

4.2.1.2. Distributed Control Schemes

Distributed control schemes use a decentralized approach to network survivability issues, where many or all network elements play some role in detecting failures and selecting the best and the quickest alternate route. The three largest categories of implementing distributed control schemes are APS, SHR and SHN [21]. However, in order to provide full restoration capabilities

APS systems require 100% redundancy, and SHRs require 100% or more redundancy. Certain SHNs, such as mesh-restorable networks, can achieve lower redundancy levels than SHRs [24].

APS comes in many flavors and is the simplest and most popular distributed control restoration mechanism. The basic premise for all APS systems relies on a set of working and a set of backup links. 1+1 (parallel) APS is based on parallel transmission of the same data on a working as well as a backup link, and, in case of a detected failure, the receiver shifts from a working link to a backup link. 1:1 (non-parallel) APS also uses one working and one backup link, only in this case nothing is being transmitted over a backup link until the failure is detected, after which both transmitter and receiver switch to the backup link. Then there is m:n APS which is just an extension of the 1:1 APS system. In order to lower some redundancy levels, generally m (the number of working links) would be smaller than n (the number of backup links). APS can easily be implemented in ATM either through means of its VPs or VCs. 1:1 VP-APS has even been discussed in ITU-T as a basic restoration function in ATM networks [21].

SHR is a high-speed restoration scheme for ring topology SONET/SDH networks and is similar to 1:1 or 1+1 APS methods. An example would be a Bi-directional Line Switched Ring (BLSR), which can be implemented either on two or four fibers and guarantees 60 ms restoration performance [22]. SHRs can also be implemented in the ATM layer, but since ATM does not depend as much on the layout of the fiber and network’s physical topology, this method does not offer any cost advantage over VP-APS.

SHN is an expansion of SHR without topological restrictions. Most commonly the best alternate route in an SHN is found by message transmission between the network nodes, often employing flooding algorithms. The clear disadvantage is that these topology update algorithms can generate many messages, and are therefore limited to restoration over small geographical areas [22].

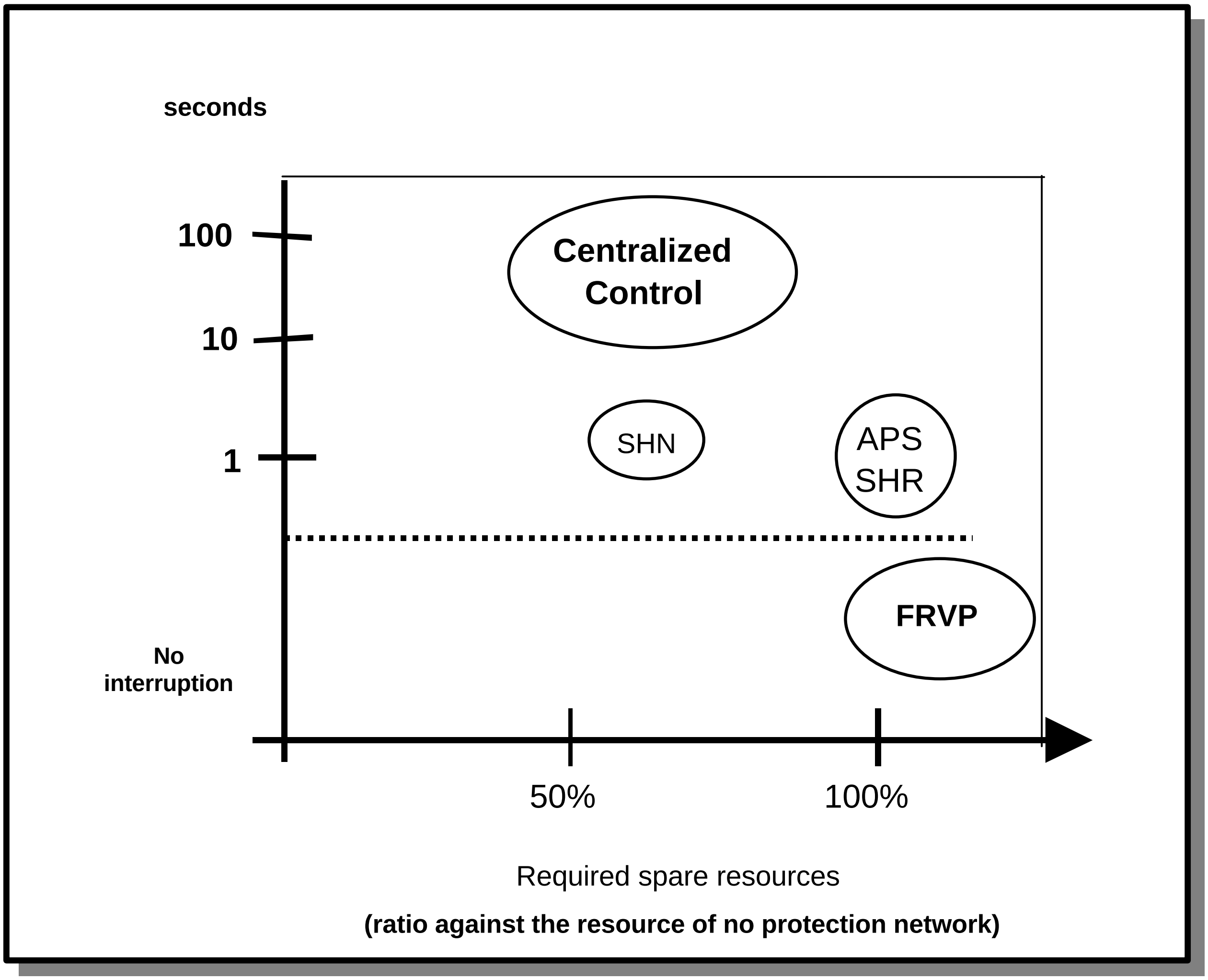


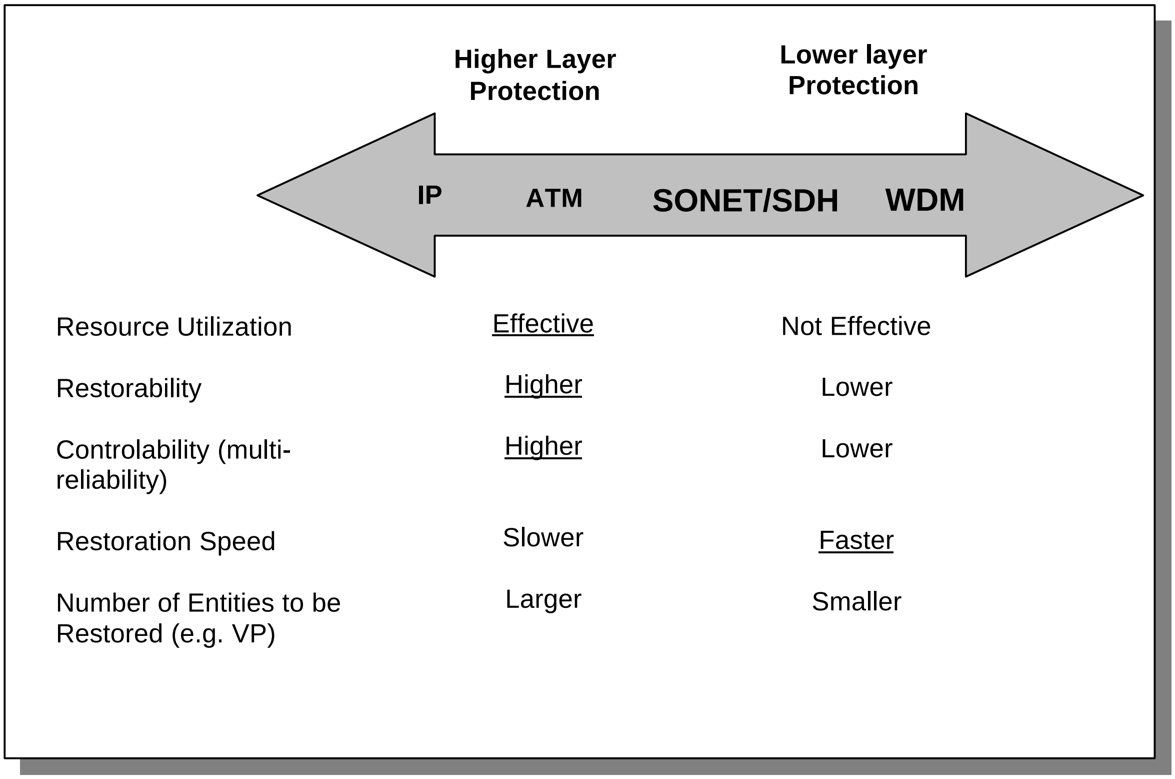
Figure 4.2 Network Restoration Schemes

FRVP realizes failure free transmission even under network failure and thus offers no interruption of service, as shown in Figure 4.2 [21]. In FRVP the transmitter would duplicate all the outgoing traffic and send it over the multiple VPs, each routed over a separate physical link. The receiver would constantly compare the cells arriving from separate VPs, choose the cells without error, and then pass the cells without error onto the user [21].

4.2.2. Multi-Layer Network Restoration

The multilayered nature of computer networks (for example, WDM-SONET-ATM-IP) makes the problem of network restoration more complex. Although each layer has its own ways of dealing with failures, and each has advantages and disadvantages (see Figure 4.3), the question becomes how efficient is the overall network restoration system. Identifying solutions to this problem of overall network protection in an efficient manner, while taking into account the individual restoration mechanisms of each layer, has been on the main agenda of projects such as ACTS' PANEL [25]. However, solutions such as layer escalation, subnetwork escalation, and scheme escalation have been offered.

Layer escalation involves a restoration process that begins at the lowest layer; if the first layer is unable to eliminate the failure, then the process proceeds layer-by-layer up the protocol stack until all the failures are removed.



	Higher Layer Protection	Lower layer Protection
Resource Utilization	Effective	Not Effective
Restorability	Higher	Lower
Controlability (multi-reliability)	Higher	Lower
Restoration Speed	Slower	Faster
Number of Entities to be Restored (e.g. VP)	Larger	Smaller

Figure 4.3 Characteristics of Each Layer

Subnetwork escalation schemes involve a restoration process starting physically close to the failed network element; if the subnetwork does not have an alternate route or additional resources, the process continues to expand the restoration until an alternate route and / or resources are found.

Scheme escalation involves restoration according to a set plan. An example would be an initial restoration attempt by a certain preplanned scheme; in the event that the plan does not eliminate all the failures, then the dynamic planned scheme would be employed [21].

4.3. Soft Failures

Soft failures are not as clear to define and have not yet been studied extensively. Soft failures can include wide variety of component or system malfunctions components where performance is compromised without triggering hard failure alarms. For example, it is possible to have several components performing at a marginal or suboptimal level where the overall performance of the system may be degraded. Therefore it is a goal of this project to dynamically load and stress the system, and to then observe its performance just before the initiation of the built in protection mechanisms, such as those previously described. The intention is to gain an insight into this area of marginal network performance where no particular layer, protocol or hardware is obviously malfunctioning, but overall performance has decreased as observed by the end user.

Chapter 5. Network Performance Metrics

5.1. Chariot

Ganymede's Chariot is a software tool designed to test end-to-end network performance of complex and multiprotocol networks, and is distributed on the PCs as end users of the network, as shown in Figure 5.1 [2].

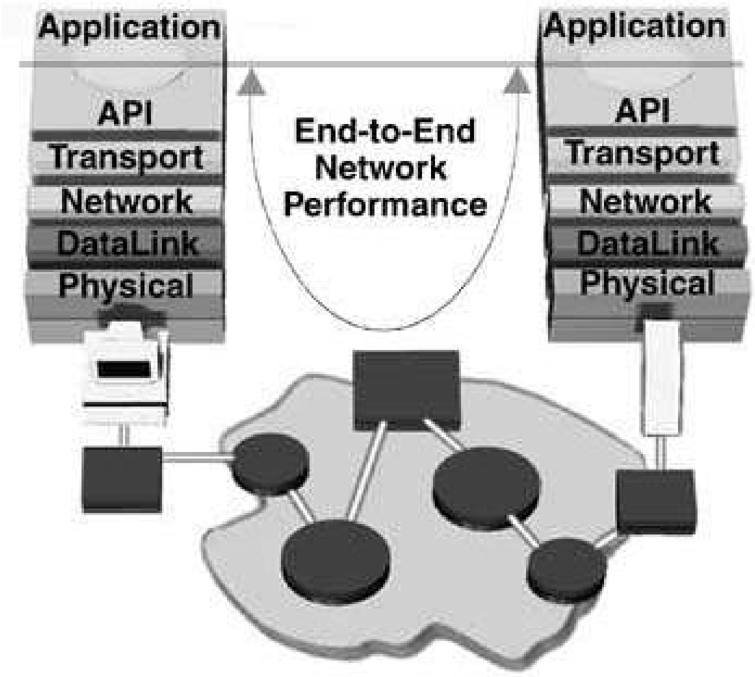


Figure 5.1 End-to-End Network Performance Measurement

There are two separate entities: 1) the Console gathers information, performs the analysis and reports the results: 2) the Network Performance Endpoints execute the Application Scripts designed to mimic actual applications running on the network [2].

An endpoint pair consists of network addresses of the endpoints, an application script which is to run between the two endpoints and the network protocol which the application scripts are to run over. These parameters are specified in the console, and once defined, the console contacts the individual endpoints of the endpoint pair. During the initialization process, the console sends the respective halves of the application script, which was previously selected to run between the endpoints, to each endpoint of the endpoint pair, and once the endpoints receive their share of application scripts the console orders the execution. Ultimately, Endpoint 1 collects the results of the test and sends the data back to the console for analysis, formatting, display and storage. Of course there could also be a multiple of different endpoint pairs running at the same time between the same two endpoints, while each pair executes a completely different script and runs on a completely different underlying protocol.

Since most of the operation of Chariot is centered around its console, it is preferable to have the console installed on a separate PC. However, due to financial constraints, the console and Endpoint 1 were placed on the same PC during the tests performed for this research.

5.1.1. Application Scripts

Chariot uses the application scripts to emulate real applications without having the need to install them. These application scripts use the same Application Programming Interfaces (APIs) for communication as do any other application such as the FTP or a Web browser [2]. Since the application scripts make the same API calls to the network protocol stack as do real applications, the protocol stack goes through the same procedures for sending and receiving data, and cannot distinguish between the APIs made from the Chariot application scripts and the real application. However, the application scripts consist of commands, which allow the user to change the parameters of the application such as the file sizes to be sent or received, type of data to be sent or received, size of the buffers, etc. In return, these script variables allow for monitoring of sent and/or received traffic and for calculation of traffic parameters. This, in turn, allows valuable insights into the network being examined.

5.1.1.1. Benchmark Scripts

Chariot v2.2 offers 12 different scripts that emulate Bader benchmarks or classic transactions, which represent most of the client/server applications. These include Credit Check, Database Update, File Transfer – Receive, File Transfer – Send, Inquiry, and Packet Blaster, which is offered as Packet Blaster (long send) and Packet Blaster Reversed (Long Receive). Each of the first five application scripts come in two options -- one using the long and one using the short connections. The main difference is that the application scripts identified as long connections use the first connection established between the endpoint pair for all transactions that occur for the duration of the script, while the scripts identified as short connections establish a new connection for each transaction.

5.1.1.2. Internet Scripts

These scripts replicate the common applications used on the Internet today. The categories include application scripts FTP Get and Put, Web Graphic and Text Request, Network News, Send and Receive E-mail, and Telnet.

5.1.1.3. Web Push Scripts

Web Push scripts model four popular Web Push applications: BackWeb, Castanet Tuner, Headliner, and PointCast Network.

5.1.1.4. Business Scripts

The set of seven scripts in this category emulates two popular business applications: Lotus Notes and SAP R/3. No Web Push and Business scripts were used in network tests and are only mentioned for completeness.

5.1.1.5. Multimedia Scripts

The Multimedia scripts are used strictly with UDP or IPX protocols. They do not require reliable delivery and they differ from the rest in that the data sent to emulate multimedia applications is sent only in one direction. Endpoint 1 (E1) sends the data to Endpoint 2 (E2), which then keeps statistics on received and lost data, which are then returned as part of the results. The Multimedia application scripts emulate MPEG audio and video streams, NetMeeting audio and video streams, NetShow applications, Real Audio, and Voice Over IP.

5.2. Response Variables

As mentioned earlier, script variables allow for monitoring of sent and/or received traffic as well as for calculation of certain response variables based on the obtained traffic parameters. The following response variables are observed and compared for different attenuation levels: throughput, transaction rate, and response time for non-multimedia scripts, and throughput, lost data and percent bytes lost for multimedia scripts. The main reason for dealing with different response variables is a different underlying protocol. Non-multimedia applications mostly run on top of the TCP protocol whereas the multimedia applications run on the UDP protocol. These differences will be pointed out for each variable.

5.2.1. Throughput

Throughput is one of the most critical variables considered in this research project. The value for throughput is calculated differently for non-multimedia and multimedia application scripts.

The throughput for non-multimedia scripts is obtained from:

$$T_h = \frac{(B_s + B_r)}{t_m} \qquad \text{Equation 5.1}$$

The throughput for a multimedia script is calculated with the following equation:

$$T_h = \frac{B_r}{t_m} \qquad \text{Equation 5.2}$$

where Bs is the number of bytes sent by Endpoint 1 of a pair, Br is the number of Bytes received by the Endpoint of a pair, and Tm is the measured time which is the sum of all timing records returned for the particular endpoint pair.

5.2.2. Transaction Rate

Transaction rate provides information on how fast transactions take place by giving its value in units of transactions per second. The following equation is used to obtain transaction rate:

$$T_r = \frac{T_c}{t_m}$$

Equation 5.3

where Tc is the transaction count and Tm is the measured time.

5.2.3. Response Time

Response time is an inverse variable from the transaction rate and it provides an information on how long it takes to complete a transaction.

$$R_t = \frac{t_m}{T_c}$$

Equation 5.4

Throughput, transaction rate and response times are all calculated for each endpoint pair running between the given endpoints. The minimum and maximum values for an individual timing record, as well as the average for all endpoint connections of the same type, are also observed and displayed.

Transaction rate and response times are not calculated for multimedia scripts since they are one-way data flows.

5.2.4. E1 Throughput

In addition to the regular throughput metrics calculated for both multimedia and non-multimedia scripts, Endpoint 1 (E1) throughput represents the throughput as viewed by the first endpoint of the endpoint pair E1 when multimedia application scripts are executed over the pair. The E1 throughput does not take into account the lost data during the script execution, and therefore may give a higher number than the main throughput variable. The E1 throughput value is calculated as:

$$E1T_h = \frac{B_s}{t_m} \qquad \text{Equation 5.5}$$

where BsE1 is bytes sent by E1, and Tm is the measured time.

For multimedia application scripts, Chariot also keeps tabs and displays the value for the total amount of bytes sent by E1, and the total amount of bytes of data received by Endpoint 2 (E2), as well as the total number and percentage of bytes lost by E2.

5.2.5. Ninety-Five Percent Confidence Interval

95% Confidence Interval is a variable that offers 95% confidence (or probability of 0.95) that the stated real average throughput is within the given range or interval. The 95% Confidence Interval, C_I, is calculated as:

$$C_I = t_{val} \cdot \frac{\sigma}{\sqrt{N\text{-}1}}$$

Equation 5.6

where σ is the standard deviation of the measured time of the timing records and N is the number of timing records. t_{val} is the statistical or Student "t" value which Chariot looks up for the number of timing records minus one. The probability density function (pdf) of the Student’s “t” approaches the Gaussian pdf as N approaches infinity [26]. Thus the 95% gets smaller as the sample size increases, all things being equal.

5.2.6. Relative Precision

Whereas the 95% Confidence Interval gives good and reliable results on the calculated average throughput for the particular application script running on an endpoint pair, it is not an accurate way to compare reliability of different application scripts. An example would be comparing the reliability of the file transfer and an inquiry scripts based on the 95% Confidence Interval. It would be difficult to make sense out of this comparison of “apples and oranges;” because, as the number of transactions computed by each script is extremely different, so will their 95% Confidence Interval values be. In order to level the playing field, there is a need for a metric that enables a reliability of network performance comparison regardless of what script was run. This metric is Relative Precision, and it is obtained by calculating the 95% confidence interval of the Measured Time for each timing record and then dividing by the average Measured Time. A “good” relative precision value is 10 or less.

5.2.7. Attenuation

The attenuation of optical power through the multimode fiber was controlled by using inline variable attenuators. Two attenuators were used for this research project as shown in Figure 5.2-- one attenuator between each pair of the transmit and receive ports on the ATM switch.

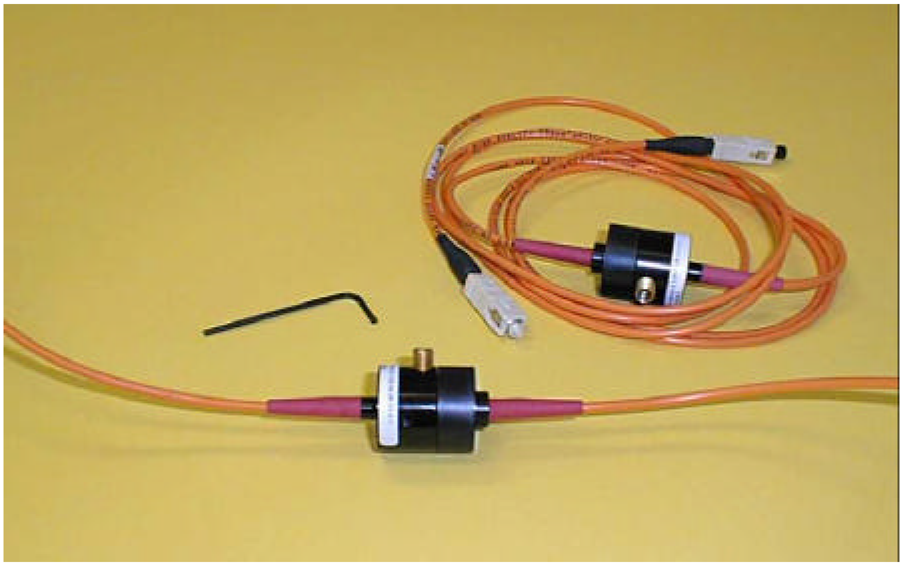


Figure 5.2 Variable Inline Attenuators Used between Ports on the ATM Switches

The principle of operation of the particular attenuator type used in this project is based on the misalignment of two fibers as shown in Figure 5.3. The top mounted screw pushes down on one of the two fiber ends creating misalignment of the fibers. This misalignment, in turn, causes light to leak out from the transmitting fiber end leaving less optical power into the receiving end, which continues down to the photo diode built into the receiving port of the ATM switch.

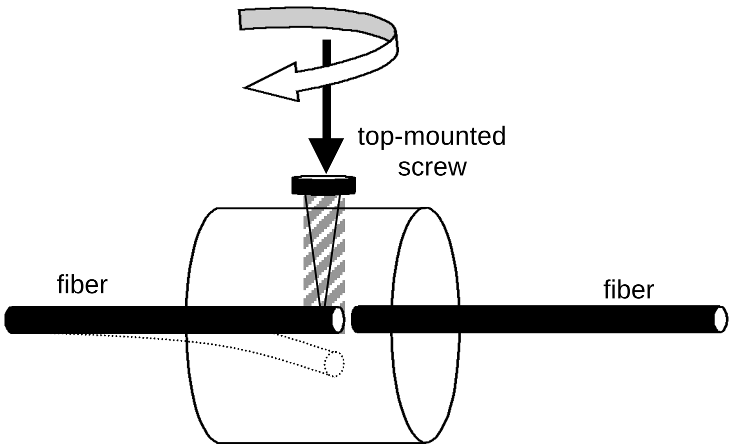


Figure 5.3 Principle of Operation of the Inline Attenuator

The attenuation values ranged between 0.8 and 49 dB as tested by the manufacturer, and the screw went through 17.75 turns. However, it was soon observed that the attenuation was not nearly linear with the turns so it was impossible to determine how much attenuation would be added or removed for each full 360 degree turn. This problem was solved by using a Tektronix TFC200 Optical Power Meter with an SC bulkhead connector already installed. The instrument was calibrated by Instrument Calibration Services in Salem on April 14, 1999 and was due for recalibration on April 14, 2000. This particular instrument is used for final product testing by FORCE Inc., in Christiansburg, VA, and was loaned by them for this project.

The optical power measured out of the switches' ports ranged from 25 µW, without any attenuation (just the insertion and coupling loss in the fiber optic jumper cable), to around 0.6 µW immediately prior to the triggering of the red indicator lights displayed on the ATM module ports as a result of insufficient optical power to the switch.

Chapter 6. Design and Implementation of the Experiment

6.1. Physical Test Setup

The test set-up included two ATM switches (Olicom CrossFire 9100 and 9200), two PCs (450 MHz Pentium III processors with 128 MB of RAM), and two NICs (RapidFire 6162 ATM 155 PCI) connected by OC-3 multimode optical fiber links. Furthermore, two variable attenuators were inserted between the two switches and the same set of tests was executed for different levels of attenuation. The photo of the actual network set-up is shown in Figure 6.1.

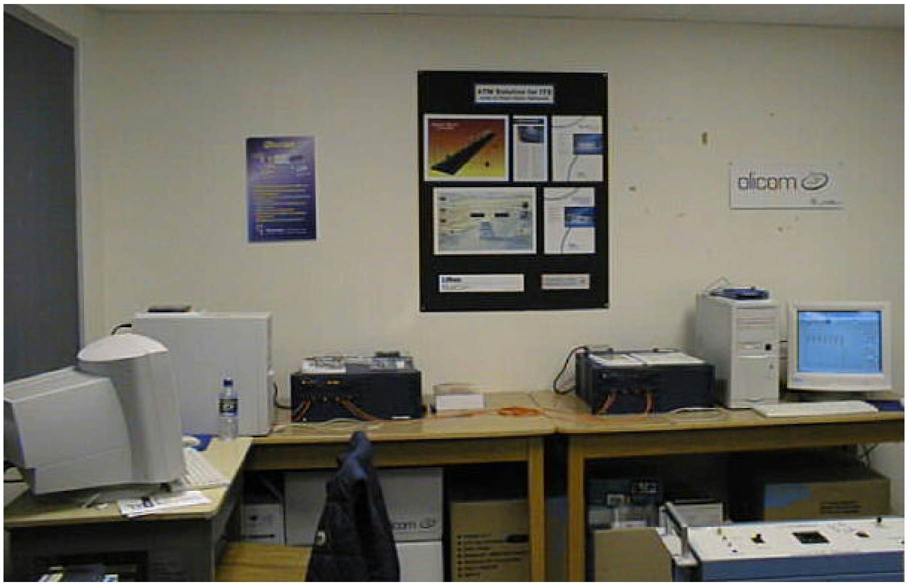


Figure 6.1 Laboratory Setup

6.2. Hardware Configuration

The ATM switches were configured to run LANE, where the 9200 operated as the primary LANE administrator and the 9100 was the secondary. The emulated protocol was Ethernet (IEEE 802.3). This configuration offered a SONET/ATM/LANE-Emulated Ethernet/IP/TCP/Application protocol stack. All hardware was given a separate LANE IP and had the same subnet mask of 255.0.0.0, as shown in Figure 6.2.

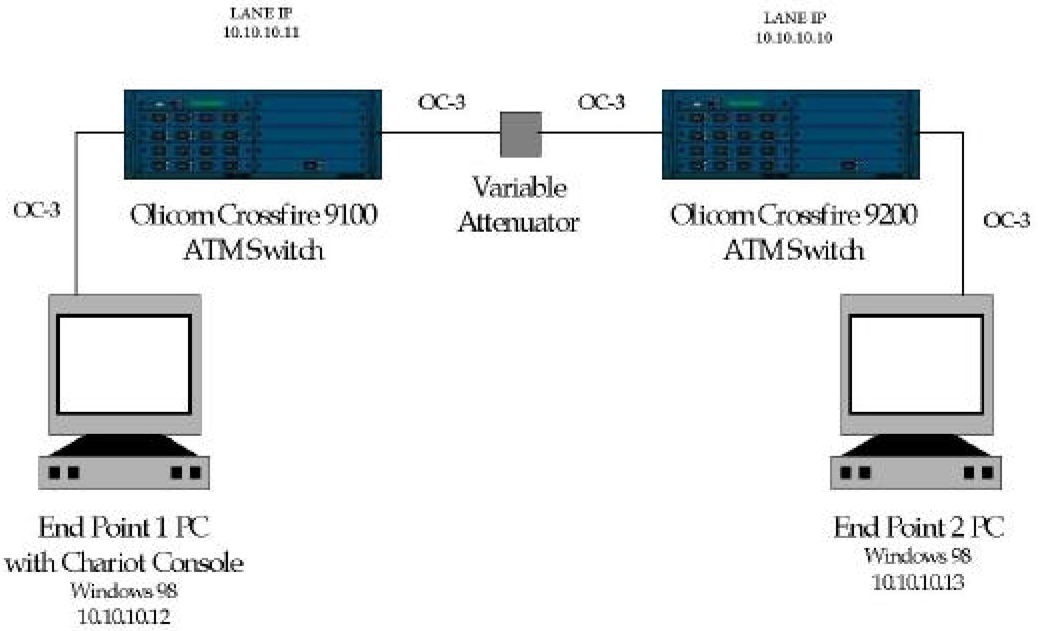


Figure 6.2 Block Diagram of Test Setup in the Laboratory

The ATM NIC's use UBR service category to relay the data traffic across the network. In the case where only the two PCs are connected to the network, the entire OC-3 bandwidth of 155.52 Mbps is available to UBR. This is an important detail since generally UBR never has all the network bandwidth at its disposal.

Configuration parameters for the PCs are presented in Table 6.1, where E1 designation refers to Endpoint 1, and E2 corresponds to Endpoint 2.

Table 6.1 PC Configuration Parameters

Endpoint 1 Type	**Endpoint 1 Value**	**Endpoint 2 Type**	**Endpoint 2 Value**
E1 Version	3.1	E2 Version	3.1
E1 Build Level	403	E2 Build Level	403
E1 Product Type	Retail	E2 Product Type	Retail
E1 Operating System	Windows 98	E2 Operating System	Windows 98
OS Version (major)	4	OS Version (major)	4
OS Version (minor)	10	OS Version (minor)	10
OS Build Number	1998	OS Build Number	1998
CSD Version		CSD Version	
Memory	130572 (KB)	Memory	130572 (KB)
APPC Default Send Size	32763	APPC Default Send Size	32763
IPX Default Send Size	1391	IPX Default Send Size	1391
SPX Default Send Size	4096	SPX Default Send Size	4096
TCP Default Send Size	4096	TCP Default Send Size	4096
UDP Default Send Size	8183	UDP Default Send Size	8183
WinSock API	Microsoft	WinSock API	Microsoft
WinSock Stack Version	2.2	WinSock Stack Version	2.2
WinSock API Version Used	2.2	WinSock API Version Used	2.2

In order to provide the best environment for obtaining consistent results, the PCs obtained for this project have identical hardware and software features, are configured in the same way, were purchased at the same time, and use the same NIC cards. The table also shows that both PCs are running identical Operating Systems (OS), Windows 98, and all corresponding parameters have the same values on both computers. The most important configuration parameters to note are the TCP send size of 406 bytes and UDP send size of 8183 bytes.

6.3. Software Configuration

The application layer and network performance tests were done by Ganymede’s Chariot software [3]. For this project, four main tests were selected and divided in two groups. Test 1 and Test 2 emulated Bader benchmark or classic transactions, and internet applications, respectively, and both used TCP as the transport layer. Test 3 and Test 4 emulated multimedia data and used UDP as the transport protocol. It is worth noting that multimedia data is generally transported over UDP protocol. The connectionless, “best effort” service of UDP and IP are necessary for effective multicast and streaming applications, which are an integral part of transporting multimedia over the Internet. Since Chariot emulates the “real world” applications, there is no option for running multimedia application scripts over TCP.

Generally, the test parameters were adjusted for achieving the highest possible throughput between the two end points. The four tests were run for a duration of one hour over nine different power levels. This duration was sufficient to obtain statistically valid data, which can be observed through the 95% Confidence Interval and Relative Precision parameters. The following parameters have been observed and compared for different attenuation levels: throughput, transaction rate, and response time for Tests 1 and 2, and throughput, percent bytes lost and number of datagrams lost between the endpoints, for Tests 3 and 4.

6.4. Testing

All test parameters were set at levels consistent with stressing the network. The goal was to cause a large amount of network traffic that would really stress the network hardware and software.

Run for a fixed duration: all tests were run for one hour, and certain script parameters were changed in order to generate a number of timing records large enough for statistical calculations, but small enough to prevent overloading the console with too many timing records.

Report timings using real-time: real-time reporting causes extra network traffic since the timing records flow across the network and are being reported as they are being generated. Batch file reporting waits until the tests are completed and then reports the results.

Regular polling of the endpoints: polling of endpoints causes additional flows outside the pattern of scripts and timing records, thus further stressing the network. In this project polling was done every minute for the duration of the test.

Validation of data upon receipt: data validation was important in order to see if there were any problems with data transferred across the network under the stress conditions. Data validation was especially important considering that the physical link was gradually degraded during this test.

Random SLEEP times: Chariot suggests using uniform distribution of sleep times to emulate many users. In this project the uniform distribution of 0 to 50 ms was used.

Setting the SEND data type to NOCOMPRESS: in addition to simply sending more data, the data that was sent is the toughest data to compress, thereby keeping any network components that use compression busy trying to find patterns in this data.

In addition to these parameters, the tests were designed to always use a new random seed when starting a new test, and the console was instructed to stop the test after the first pair failed.

6.4.1. Test One

Test one consisted of 4 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 115 Mbps. The test used two of the benchmark scripts. Two pairs ran FILERCV scripts which emulated a request for a file and getting it back. The other two pairs ran FILESND scripts which emulate sending a file across the network and then getting a confirmation back.

The run options for Test 1 were chosen according to guidelines described at the beginning of Section 6.4 and are presented in Table 6.2. In addition to the duration, reporting type, polling, and data validation it shows that the test was set to stop if there was an initialization failure between any of the four pairs, regardless of when it happened during the test. It also shows that a new seed for random variables was used for every run.

Table 6.2 Test 1 Configuration Parameters

End type	**Run for a fixed duration**
Duration	01:00:00
Reporting type	Real-time
Automatically poll endpoints	Yes
Polling interval (minutes)	1
Stop run upon initialization failure	Yes
Connect timeout during test (minutes)	0
Stop test after this many running pairs fail	1
Validate data upon receipt	Yes
Use a new seed for random variables on every run	Yes

The details specifying the pairs and their respective endpoint's addresses, application scripts, and the network protocol used for the test setup, are shown in Table 6.3. It should be mentioned that Chariot uses only TCP to communicate with Endpoint 1, regardless of which network protocol is being used by the applications scripts between the endpoints.

Table 6.3 Test 1 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	**Endpoint 1**	**Endpoint 2**	**Network Protocol**	**Service Quality**	**Script Name**
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	TCP		filercvltest1.scr
Pair 2	10.10.10.12	10.10.10.13	TCP		filercvltest1.scr
Pair 3	10.10.10.12	10.10.10.13	TCP		filesndltest1.scr
Pair 4	10.10.10.12	10.10.10.13	TCP		filesndltest1.scr

6.4.1.1. File Receive, Long Connection Script

As can be seen in Table 6.3, the first two pairs ran the filercvltest1.scr script, which performs the same function as the FILERCVL.SCR but with a few changed parameters in order to provide the higher throughput. The changed script was named *filercvltest1* and it is presented in Table 6.4.

Table 6.4 filercvltest1.scr Script

ENDPOINT 1	**ENDPOINT 2**
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
Number of timing records=100	Number of timing records=100
START TIMER	
LOOP	LOOP
transactions_per_record=1	Transactions per record=1
SEND	RECEIVE
size_of_record_to_send=100	size_of_record_to_send=100
size_of_record_to_send=100	size_of_record_to_send=100
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
RECEIVE	SEND
file_size=10000000	file_size=10000000
receive_buffer_size=DEFAULT	send_buffer_size=DEFAULT
	send_datatype=NOCOMPRESS
	send_data_rate=UNLIMITED
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
END_TIMER	
SLEEP	
transaction_delay=0	
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Following the guidelines, and in order to maximize the throughput, a few script variables were changed: Initial delay was set to have uniform distribution values between 0 and 50 ms, and the file size was increased. The detailed description of the script parameters, their values, and descriptions are given in Table 6.5.

Table 6.5 Description of Variables for filercvltest1.scr Script

Variable Name	**Value**	**Description**
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	100	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	100	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
transaction_delay	0	Milliseconds to pause
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.1.2. File Send, Long Connection Script

The second two pairs ran the filesndltest1.scr script, which performs the same function as the FILESNDL.SCR but with a few changed parameters in order to provide the higher throughput. The changed script was named *filesndltest1* and it is presented in Table 6.6, and the individual description of parameters used in the script can be found in Table 6.7.

Table 6.6 filesndtest1.scr Script

ENDPOINT 1	ENDPOINT 2
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=10	number_of_timing_records=10
START_TIMER	
LOOP	LOOP
transactions_per_record=1	transactions_per_record=1
SEND	RECEIVE
file_size=10000000	file_size=10000000
send_buffer_size=DEFAULT	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
CONFIRM_REQUEST	CONFIRM_ACKNOWLEDGE
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
END_TIMER	
SLEEP	
transaction_delay=0	
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.7 Description of Variables for filesndtest1.scr Script

VARIABLE NAME	VALUE	DESCRIPTION
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	10	How many timing records to generate
transactions_per_record	1	Transactions per timing record
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
transaction_delay	0	Milliseconds to pause
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.2 Test Two

Test 2 consisted of 3 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 108 Mbps. The test used three of the Internet scripts, which were slightly modified for this project: FTPPUT, HTTPGIF and HTTPTEXT.

Since both Test 1 and Test 2 run over TCP, the run options for this test template are the same as presented in Table 6.2. The details for the test setup from Endpoint 1 to Endpoint 2, such as the network protocol and the script name for the group of pairs 1, 2, and 3, are presented in Table 6.8.

Table 6.8 Test Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	TCP		*ftpputtest2.scr*
Pair 2	10.10.10.12	10.10.10.13	TCP		*httpgiftest2.scr*
Pair 3	10.10.10.12	10.10.10.13	TCP		*httptexttest2.scr*

6.4.2.1. FTP Put Script

The first pair ran an FTPPUT script, which emulated the TCP/IP's FTP application, and where Endpoint 1 is placing a file onto Endpoint 2. The modified FTPPUT script, ftpputtest2, is shown in Table 6.9, and detailed descriptions of the changed variables are presented in Table 6.10.

Table 6.9 ftpputtest2.scr Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=u[0,50]	
LOOP	LOOP
number_of_repetitions=1	number_of_repetitions=1
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=auto	port_number=auto
RECEIVE	SEND
login_size=15	login_size=15
control_buffer_size=DEFAULT	control_buffer_size=DEFAULT
	......control_datatype=trans.cmp
	send_data_rate=UNLIMITED
SLEEP	
user_delay=0	
SEND	RECEIVE
login_size=15	login_size=15
control_buffer_size=DEFAULT	control_buffer_size=DEFAULT
control_datatype=trans.cmp	
send_data_rate=UNLIMITED	
RECEIVE	SEND
login_size=15	login_size=15
control_buffer_size=DEFAULT	control_buffer_size=DEFAULT
	control_datatype=trans.cmp
	send_data_rate=UNLIMITED
SLEEP	
user_delay=0	
SEND	RECEIVE
login_size=15	login_size=15
control_buffer_size=DEFAULT	control_buffer_size=DEFAULT
control_datatype=trans.cmp	
send_data_rate=UNLIMITED	
RECEIVE	SEND
200	200
control_buffer_size=DEFAULT	200
	control_datatype=trans.cmp
	send_data_rate=UNLIMITED
DISCONNECT	DISCONNECT
LOOP	LOOP
number_of_timing_records=100	number_of_timing_records=100
START TIMER	
LOOP	LOOP
transactions_per_record=1	transactions_per_record=1
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
SEND	RECEIVE
file_control_size=30	file_control_size=30
control_buffer_size=30	control_buffer_size=30
control_datatype=trans.cmp	

```
      send_data_rate=UNLIMITED
    RECEIVE
      file_control_size=30
        control_buffer_size=DEFAULT


      SEND
        file_control_size=30
        control_buffer_size=DEFAULT
        control_datatype=trans.cmp
        send_data_rate=UNLIMITED
      RECEIVE
        file_control_size=30
        control_buffer_size=DEFAULT


      SLEEP
        delay_before_responding=0
      SEND
        size_of_record_to_send=10000000
        send_buffer_size_4096
        send_datatype=NOCOMPRESS
        Send_data_rate=UNLIMITED
      RECEIVE
        file_control_size=30
        control_buffer_size=DEFAULT


      DISCONNECT
      INCREMENT_TRANSACTION
    END_LOOP
    END_TIMER
    SLEEP
      transaction_delay=0
  END_LOOP
  CONNECT_INITIATE
   port_number=AUTO
  SEND
    6
    6
    control_datatype=trans.cmp
    send_data_rate=UNLIMITED
  RECEIVE
    25
    control_buffer_size=DEFAULT


  DISCONNECT
END_LOOP
```

```

    SEND
      file_control_size=30
        control_buffer_size=DEFAULT
        control_datatype=trans.cmp
        send_data_rate=UNLIMITED
      RECEIVE
        file_control_size=30
        control_buffer_size=DEFAULT


      SEND
        file_control_size=30
       control_buffer_size=DEFAULT
        control_datatype=trans.cmp
        send_data_rate=UNLIMITED


      RECEIVE
        file_control_size=30
        control_buffer_size=DEFAULT


     SEND
        file_control_size=30
        control_buffer_size=DEFAULT
        control_datatype=trans.cmp
        send_data_rate=UNLIMITED
      DISCONNECT

    END_LOOP



  END_LOOP
  CONNECT_ACCEPT
    port_number=AUTO
  RECEIVE
    6
    control_buffer_size=DEFAULT


  SEND
    25
    25
    control_datatype=trans.cmp
    send_data_rate=UNLIMITED
  DISCONNECT
END_LOOP
```

Table 6.10 Description of Variables for ftpputtest2.scr Script

VARIABLE NAME	VALUE	DESCRIPTION
initial_delay	u[0, 50]	Pause before the first transaction
number_of_repetitions	1	How many times to repeat the script
number_of_timing_records	100	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	10000000	Amount of data to be sent
user_delay	0	Pause before answering
transaction_delay	0	Milliseconds to pause
delay_before_responding	0	Milliseconds to wait before responding
file_control_size	30	How many bytes are in the control flows
login_size	15	How many bytes are in the login flows
control_buffer_size	DEFAULT	Buffer size for control flows
send_buffer_size	4096	How many bytes of data in each SEND
receive_buffer_size	4096	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.2.2. HTTP GIF Script

The second pair ran an HTTPGIF script, which emulates a transfer of an image in a GIF format from an HTTP server. The modified HTTPGIF script, httpgiftest2, is shown in Table 6.11, and detailed descriptions of the changed variables are presented in Table 6.12.

Table 6.11 httptest2.scr Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=0	
LOOP	LOOP
number_of_timing_records=50	number_of_timing_records=50
START_TIMER	
LOOP	LOOP
transactions_per_record=1	transactions_per_record=1
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
SEND	RECEIVE
size_of_record_to_send=300	size_of_record_to_send=300
size_of_record_to_send=300	size_of_record_to_send=300
control_datatype=trans.cmp	
send_data_rate=UNLIMITED	
	SLEEP
	delay_before_responding=0
RECEIVE	SEND
file_size=10000000	file_size=10000000
receive_buffer_size=DEFAULT	send_buffer_size=DEFAULT
	send_datatype=lena.cmp
	send_data_rate=UNLIMITED
DISCONNECT	DISCONNECT
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
END_TIMER	
SLEEP	
transaction_delay=0	
END_LOOP	END_LOOP

Table 6.12 Description of Variables for httpgiftest2.scr Script

Variable Name	**Value**	**Description**
initial_delay	0	Pause before the first transaction
number_of_timing_records	50	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	300	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
delay_before_responding	0	Milliseconds to wait before responding
transaction_delay	0	Milliseconds to pause
send_datatype	lena.cmp	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endp

6.4.2.3. HTTP Text Script

The third one ran an HTTPTEXT script, which emulates a transfer of text files from an HTTP server. The modified HTTPTEXT script, httptexttest2, is shown in Table 6.13, and detailed descriptions of the changed variables are presented in Table 6.14.

Table 6.13 httptexttest2.scr Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=u[0,50]	
LOOP	LOOP
number_of_timing_records=50	number_of_timing_records=50
START_TIMER	
LOOP	LOOP
transactions_per_record=1	transactions_per_record=1
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
SEND	RECEIVE
size_of_record_to_send=300	size_of_record_to_send=300
size_of_record_to_send=300	size_of_record_to_send=300
control_datatype=trans.cmp	
send_data_rate=UNLIMITED	
	SLEEP
	delay_before_responding=0
RECEIVE	SEND
file_size=10000000	file_size=10000000
receive_buffer_size=DEFAULT	send_buffer_size=DEFAULT
	send_datatype=news.cmp
	send_data_rate=UNLIMITED
DISCONNECT	DISCONNECT
INCREMENT_TRANSACTION	
END_LOOP	END_LOOP
END_TIMER	
SLEEP	
transaction_delay=0	
END_LOOP	END_LOOP

Table 6.14 Description of Variables for httptexttest2.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	50	How many timing records to generate
transactions_per_record	1	Transactions per timing record
size_of_record_to_send	300	Amount of data to be sent
file_size	10000000	How many bytes in the transferred file
send_buffer_size	DEFAULT	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
delay_before_responding	0	Milliseconds to wait before responding
transaction_delay	0	Milliseconds to pause
send_datatype	news.cmp	What type of data to send
control_datatype	trans.cmp	What type of control data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3. Test Three

Test three consisted of 10 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 14 Mbps. The test used six multimedia scripts: There were 5 pairs emulating Microsoft's NetMeeting Audio, Microsoft's NetMeeting Video, Microsoft's NetShow application, RealAudio by RealNetworks, and RealMedia by RealNetworks. The other five pairs all ran Voice-Over-IP (VOIP) scripts. All of the applications ran over UDP.

Test 3, with ten pairs running simultaneously, was extremely CPU intensive and was unable to generate significant overall throughput between the endpoints. However, it did show that the attenuation changes do not have much effect on network performance if the network is significantly underutilized.

The configuration parameters for Test 3 were also chosen according to guidelines described at the beginning of Section 6.4, and are presented in Table

6.15. In addition to the options such as duration, reporting type, polling, data validation, stop test after initialization failure, and new seed for random variables -- which are the same run options on all tests -- scripts running over UDP must specify the datagram window size, datagram retransmission time-out, datagram number of retransmissions before aborting the test, receive time-out, and time-to-live. Time-to-live was able to be set at one hop due to the fact that all network elements used in this project were on the same subnet.

Table 6.15 Test 3 Configuration Parameters

End type	**Run for a fixed duration**
Duration	01:00:00
Reporting type	Real-time
Automatically poll endpoints	Yes
Polling interval (minutes)	1
Stop run upon initialization failure	Yes
Connect timeout during test (minutes)	0
Stop test after this many running pairs fail	1
Validate data upon receipt	Yes
Use a new seed for random variables on every run	Yes
Datagram window size (bytes)	1500000
Datagram retransmission timeout (milliseconds)	200
Datagram number of retransmits before aborting	50
Receive Timeout (milliseconds)	10000
Time To Live (Hops)	1

The details specifying the pairs and their respective endpoints addresses, applications scripts that were used, and network protocol used for the test are shown in Table 6.16.

Table 6.16 Test 3 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	UDP		netmtgatest3.scr
Pair 2	10.10.10.12	10.10.10.13	UDP		netmtgvtest3.scr
Pair 3	10.10.10.12	10.10.10.13	UDP		netshowutest3.scr
Pair 4	10.10.10.12	10.10.10.13	UDP		realaudtest3.scr
Pair 5	10.10.10.12	10.10.10.13	UDP		realmedtest3.scr
Pair 6	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 7	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 8	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 9	10.10.10.12	10.10.10.13	UDP		voipstest3.scr
Pair 10	10.10.10.12	10.10.10.13	UDP		voipstest3.scr

6.4.3.1. NetMeeting Audio Stream Script

The first pair ran a NETMTGA script, which emulates sending an audio stream, using Microsoft’s NetMeeting Version 2.1, over a 100Mbps Ethernet LAN. The modified script, netmtgatest3, is shown in Table 6.17.

Table 6.17 netmtgatest3.scr Script

Endpoint 1	Endpoint 2
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=1760000	file_size=1760000
send_buffer_size=44	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Detailed descriptions of the variables used in this script and their values are presented in Table 6.18.

Table 6.18 Description of Variables of netmtgatest3.scr Script

Variable Name	**Value**	**Description**
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	1760000	How many bytes in the transferred file
send_buffer_size	44	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.2. NetMeeting Video Stream Script

The second pair ran a NETMTGV script, which emulates sending a video stream, using Microsoft's NetMeeting Version 2.1, over a 100Mbps Ethernet LAN. The modified NETMTGV script, netmtgvtest3, is shown in Table 6.19.

Table 6.19 netmtgvtest3.scr Script

ENDPOINT 1	**ENDPOINT 2**
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=20880000	file_size=20880000
send_buffer_size=522	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Descriptions of the variables used in the script are presented in Table 6.20.

Table 6.20 Description of Variables for netmtgvtest3.scr Script

Variable Name	**Value**	**Description**
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	20880000	How many bytes in the transferred file
send_buffer_size	522	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.3. NetShow Script

The third pair ran a NETSHOW script, which emulated Microsoft's NetShow application. The modified NETSHOW script, netshowtest3, is presented in Table 6.20, and detailed descriptions of the variables used in the script are presented in Table 6.21.

Table 6.21 netshowutest.scr3 Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=2	number_of_timing_records=2
	START_TIMER
SEND	RECEIVE
file_size=10520000	file_size=10520000
send_buffer_size=526	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.22 Description of Variables for netshowutest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	2	How many timing records to generate
file_size	10520000	How many bytes in the transferred file
send_buffer_size	526	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.4. RealAudio CD Streaming Script

The fourth pair ran a REALAUD script, which emulates the Real Audio application by RealNetworks, such as CD streaming. The modified REALAUD script, realaudtest3, is presented in Table 6.23, and detailed descriptions of the variables used in the script are presented in Table 6.24.

Table 6.23 realaudtest3.scr Script

Endpoint 1	Endpoint 2
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=14040000	file_size=14040000
send_buffer_size=351	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.24 Description of variables for realaudtest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	14040000	How many bytes in the transferred file
send_buffer_size	351	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.5. RealMedia Stream Smart Script

The fourth pair ran a REALMED script, which emulates the Real Audio application by RealNetworks, such as their "smart" audio-video stream. The modified REALMED script, realmedtest3, is presented in Table 6.25, and detailed descriptions of the variables used in the script are presented in Table 6.26.

Table 6.25 realmedtest3.scr Script

Endpoint 1	Endpoint 2
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=17240000	file_size=17240000
send_buffer_size=431	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.26 Description of Variables for realmedtest3.scr Script

Variable Name	Value	Description
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	17240000	How many bytes in the transferred file
send_buffer_size	431	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.3.6. Voice-Over-IP Script

VOIPS script, which emulates voice-over-IP application, simultaneously ran over pairs 6, 7, 8, 9 and 10. This script is presented in Table 6.27, and descriptions of the variables used in the script are given in Table 6.28.

Table 6.27 voipstest3.scr Script

Endpoint 1	Endpoint 2
SLEEP	
initial_delay=0	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=456000	file_size=456000
send_buffer_size=40	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.28 Descriptions of Variables for voipstest3.scr Script

Variable Name	Value	Description
initial_delay	0	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	456000	How many bytes in the transferred file
send_buffer_size	40	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.4. Test Four

Test 4 also consisted of 4 pairs running simultaneously across the network from Endpoint 1 to Endpoint 2 for a sustained overall throughput of around 120 Mbps. Two pairs were running two MPEG video applications. The other two pairs were running two MPEG audio scripts. Test 4 applications ran over UDP, which exhibited the most sensitivity to soft failures, and thus became the most significant test for this project. Since this test used UDP as the network protocol, as was the case in Test 3, the configuration parameters are the same as those in Table 6.15. The details about the Test 4 setup, including the application scripts used, are presented in Table 6.29.

Table 6.29 Test 4 Setup from Endpoint 1 to Endpoint 2

Group/ Pair	Endpoint 1	Endpoint 2	Network Protocol	Service Quality	Script Name
All Pairs					
Pair 1	10.10.10.12	10.10.10.13	UDP		iptvvstress.scr
Pair 2	10.10.10.12	10.10.10.13	UDP		iptvvstress.scr
Pair 3	10.10.10.12	10.10.10.13	UDP		iptvastress.scr
Pair 4	10.10.10.12	10.10.10.13	UDP		iptvastress.scr

6.4.4.1. IP/TV MPEG Video Stream Script

Pairs 1 and 2 ran IPTVV script, which emulates Cisco System's application for running an MPEG video stream over IP. It is presented in Table 6.30, and descriptions of the variables used are given in Table 6.31.

Table 6.30 iptvvstress.scr Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
SEND	START_TIMER
file_size=7300000	RECEIVE
send_buffer_size=18250	file_size=7300000
send_datatype=NOCOMPRESS	receive_buffer_size=DEFAULT
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.31 Description of Variables for iptvvstress.scr Script

Variable Name	**Value**	**Description**
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	7300000	How many bytes in the transferred file
send_buffer_size	18250	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

6.4.4.2. IP/TV Audio Stream Script

Pairs 3 and 4 ran IPTVA script, which emulates Cisco System's application for running an MPEG audio stream over IP. The script used is presented in Table 6.32, and descriptions of the variables used are given in Table 6.33.

Table 6.32 iptvastress.scr Script

Endpoint 1	**Endpoint 2**
SLEEP	
initial_delay=u[0,50]	
CONNECT_INITIATE	CONNECT_ACCEPT
port_number=AUTO	port_number=AUTO
LOOP	LOOP
number_of_timing_records=1	number_of_timing_records=1
	START_TIMER
SEND	RECEIVE
file_size=5112000	file_size=5112000
send_buffer_size=12780	receive_buffer_size=DEFAULT
send_datatype=NOCOMPRESS	
send_data_rate=UNLIMITED	
	END_TIMER
END_LOOP	END_LOOP
DISCONNECT	DISCONNECT

Table 6.33 Description of Variables for iptvastress.scr Script

Variable Name	**Value**	**Description**
initial_delay	u[0, 50]	Pause before the first transaction
number_of_timing_records	1	How many timing records to generate
file_size	5112000	How many bytes in the transferred file
send_buffer_size	12780	How many bytes of data in each SEND
receive_buffer_size	DEFAULT	How many bytes of data in each RECEIVE
send_datatype	NOCOMPRESS	What type of data to send
send_data_rate	UNLIMITED	How fast to send data
port_number	AUTO	What port to use between endpoints

Chapter 7. Results

Network performance was evaluated for each of the four test templates described in Chapter 6 based on the values for throughput, transaction rate, and response time for Tests 1 and 2, and throughput, percent bytes lost and number of datagrams lost between the endpoints for Tests 3 and 4. Each performance metric was measured and evaluated for different optical power levels. Finally, the Bit Error Rate (BER) characterization of the fiber links, including the attenuators, was evaluated and the results are presented at the end of this chapter in Section 7.5.

7.1. Test 1 Results

This section gives a typical trace of a particular metric for scripts *filercvltest1* and *filesndltest1* running over TCP, from one test run at a certain power level, and then a graph representing all the average values obtained at different optical power levels.

7.1.1. Test 1 Throughput

The typical trace for the overall throughput of all four pairs over the one-hour test duration is shown in Figure 7.1. Pairs one and two, running the same *filercvltest1* script, averaged about 29 Mbps each; pairs 3 and 4, which ran *filesndltest1*, achieved about 28 Mbps. The average overall throughput for all four pairs at this particular optical power level was 115.010 Mbps.

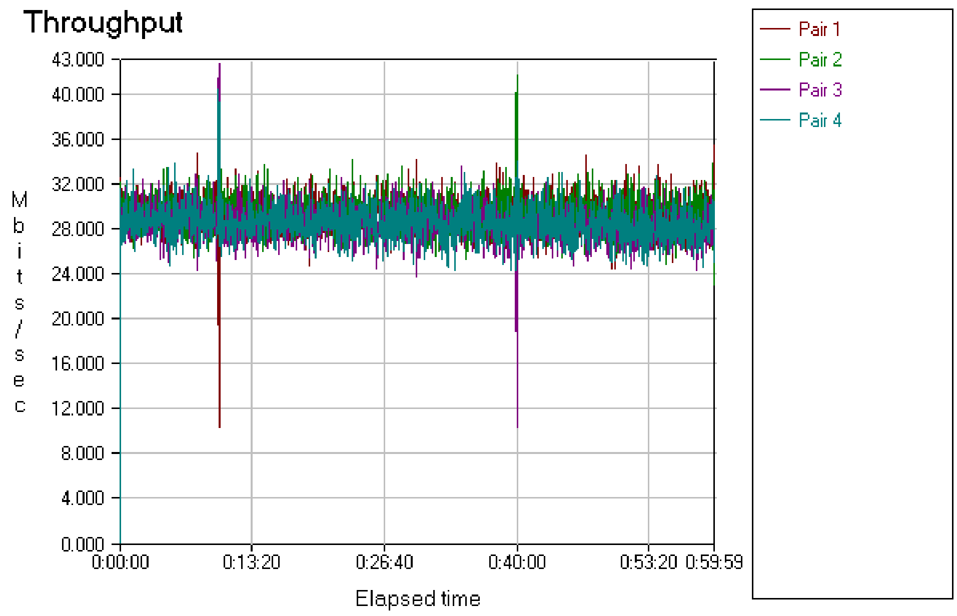
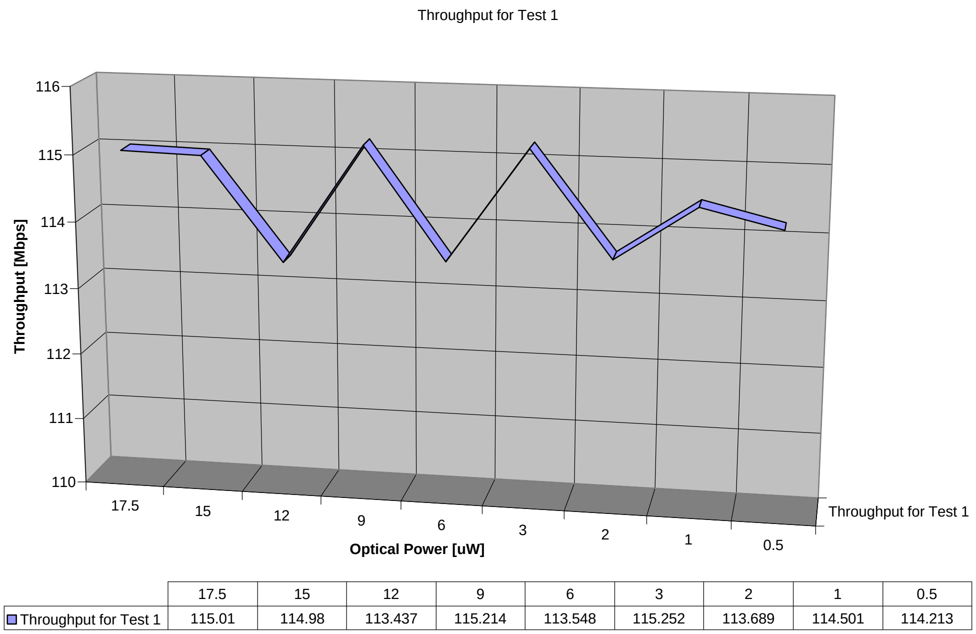


Figure 7.1 Test 1 Throughput at 17.5 mW of Optical Power

The same test run was repeated eight more times at different settings of the attenuators, and these values are given in Figure 7.2. From the figure, it can be observed that the average throughput has a peculiar periodic shape over most of the operating power region. The strange oscillation in throughput may be related to two main attributes of this test setup: 1) the removal occurs soon after the light from the transmitter in the ATM port launches light into the fiber, 2) there is a particular way that the attenuators remove optical power from the link.

The first factor has to do with the nature of light propagation through a multimode fiber. Light from a Light Emitting Diode (LED) is coupled into a relatively large core of the multimode fiber (62.5 µm diameter versus 9 µm for a singlemode fiber), where it excites a large number of different modes of propagation. These modes are the Bessel functions, which are the multiple solutions of Maxwell's equation for a cylindrical waveguide.



	17.5	15	12	9	6	3	2	1	0.5
▨ Throughput for Test 1	115.01	114.98	113.437	115.214	113.548	115.252	113.689	114.501	114.213

Figure 7.2 Test 1 Average Throughput for Different Optical Power Levels

The particular Bessel functions or modes that propagate through the fiber are decided based on the value of the boundary conditions or the actual size of the core of the fiber. Thus, the size of a singlemode fiber is designed to allow only one mode of propagation, while the larger size of multimode fiber allows for optical power, or a signal, to be transported through the fiber over a large number of different modes. The normalized propagation constant corresponding to a certain mode describes how the mode propagates through the fiber, as well as which modes die out fast. In addition to certain modes dying out fast, the peculiarities of the fiber will cause certain modes to couple into the cladding of the fiber and certain ones to completely couple out of the fiber. Generally, when the length of the fiber is sufficient, a cleansing of “dying” or “leaving” modes occurs, as well as an averaging of the “surviving” modes. However, the short distance associated with fibers replacing existing copper cabling in today’s

LANs, and especially the fibers used in this particular test (all fiber links were either 1m or 2 m in length), are not long enough to provide for mode "ironing."

The second factor, coupled with the different propagating modes in the first few meters of the fiber, is the attenuators used for this setup. The attenuators remove optical power from the fiber by misaligning the fibers, as described in Chapter 5, Section 5.2.7. Unfortunately, with only 1 m of fiber between the attenuators and the LED, the attenuators also act as very powerful mode filters. Although multimode fibers, modal filters and modal as well as other types of noise in fibers, have been studied extensively during the 80's and the 90's, characterizing the light propagation through multimode fiber remains a pretty difficult task. The task becomes even more difficult when attempting to study the light during its initial period of propagation through the multimode fiber.

Since there is a great possibility that certain modes are selectively dropped while certain ones may be reflected back from the interface, these two factors may have to do with the oscillations in performance. The overall trend of decreasing throughput, especially below the 3 μW power level has to do, as expected, with lower optical power level, which decreases the signal-to-noise ratio and ultimately increases the BER.

7.1.2. Test 1 Transaction Rate

The typical trace for the transaction rate of all four pairs over the one-hour test duration is shown in Figure 7.3. The transaction rate provides information on how fast transactions between the two endpoints take place. The units are in number of transactions per second.

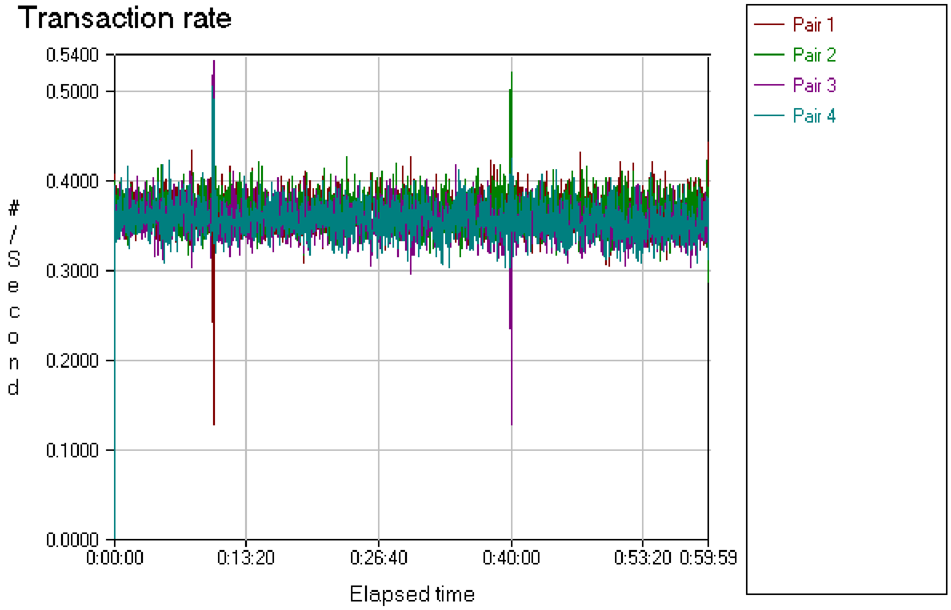
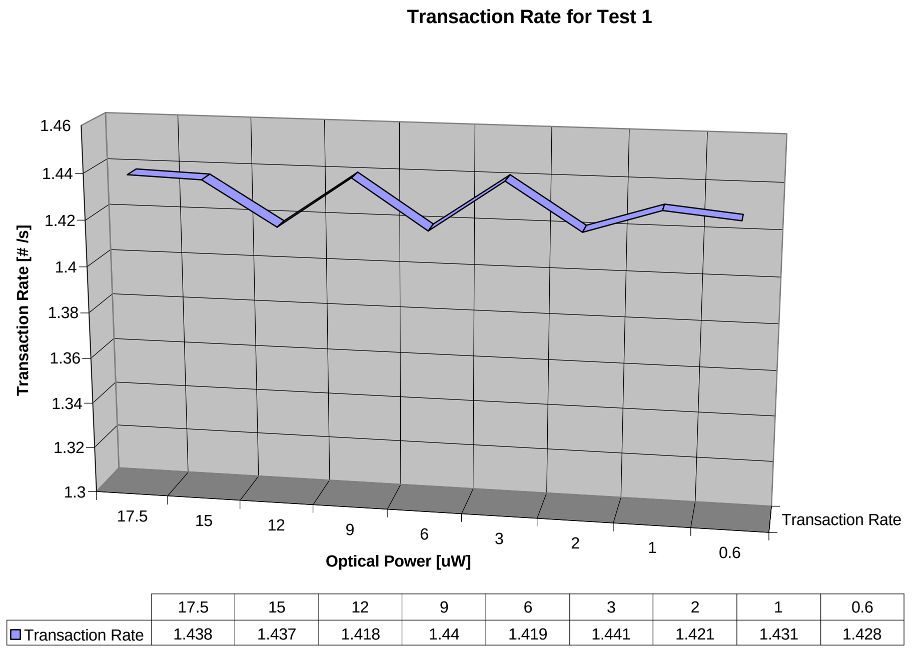


Figure 7.3 Test 1 Transaction Rate at 17.5 mW of Optical Power

Pairs 1 and 2 achieved around 0.36 transactions per second, whereas pairs 3 and 4 achieved about 0.35 transactions per second. The overall transaction rate average for this power level was around 1.44.

Transaction rate values at different settings of the attenuators are given in Figure 7.4. As expected, the shape of the transaction rate graph closely resembles the throughput graph, both in its oscillatory variations over the evaluated power range and in the overall-decreasing trend below the 3 µW level.



	17.5	15	12	9	6	3	2	1	0.6
▨ Transaction Rate	1.438	1.437	1.418	1.44	1.419	1.441	1.421	1.431	1.428

Figure 7.4 Test 1 Transaction Rate for Different Optical Power Levels

7.1.3. Test 1 Response Time

The response time of all four pairs over the one-hour test duration is presented in Figure 7.5. Response time is an inverse variable from the transaction rate, and it provides information on the length of time it takes to complete a transaction. Pairs 1 and 2 needed around 2.74 seconds per transaction, whereas Pairs 3 and 4 needed about 2.82 seconds per transaction. The overall response time average for this power level was around 2.78.

Response time values at different optical power levels are given in Figure 7.6. The graph clearly shows that the response time parameter is the inverse of the transaction rate. It also shows that the time it takes to complete a transaction increases at lower power levels, again especially below 3μW of power.

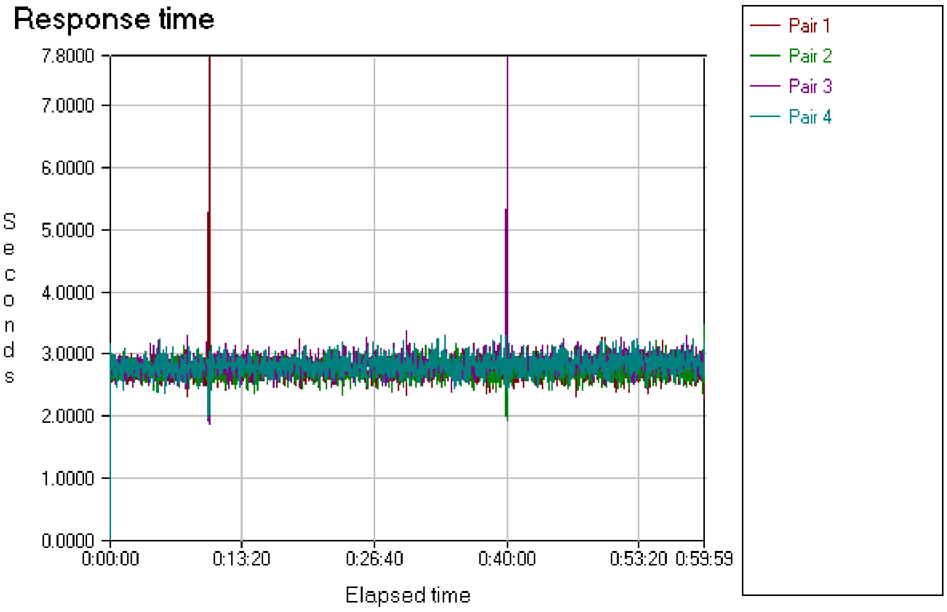
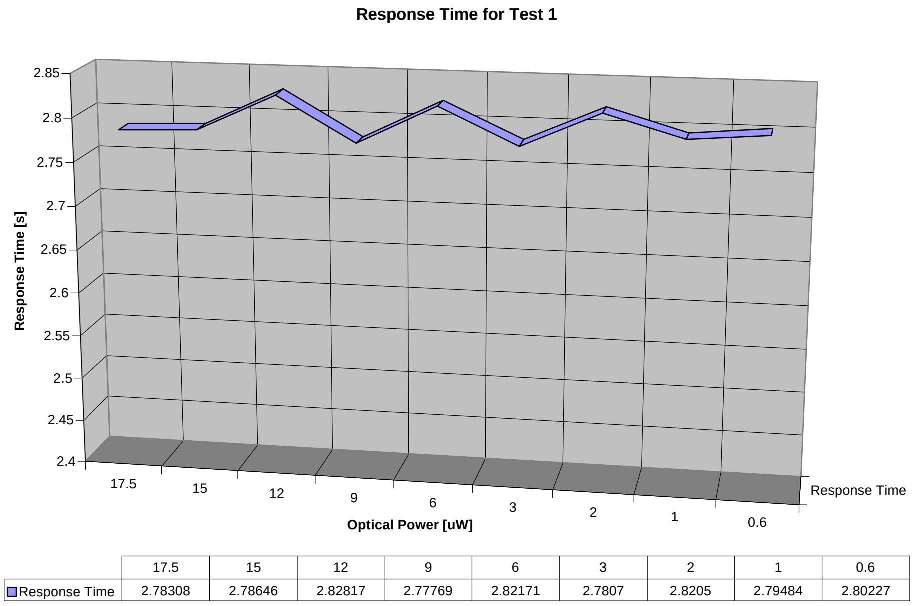


Figure 7.5 Test 1 Response Time at 17.5 mW of Optical Power



	17.5	15	12	9	6	3	2	1	0.6
▨ Response Time	2.78308	2.78646	2.82817	2.77769	2.82171	2.7807	2.8205	2.79484	2.80227

Figure 7.6 Test 1 Response Time for Different Optical Power Levels

7.2. Test 2 Results

This section gives a typical trace of a particular metric for *ftpputtest2, httpgiftest2* and *httptexttest2* scripts running over TCP. One test was run at a certain power level, and then the graph was generated showing the average values obtained at different optical power levels.

7.2.1. Test 2 Throughput

The typical trace for the overall throughput of all four pairs over the one-hour test duration is shown in Figure 7.7. Pair 1 running the *ftpputtest2* script achieved about 34.9 Mbps; pair 2 running the script *httpgiftest2* achieved about 36.9 Mbps; and pair 3 running the *httptexttest2* achieved about 36.6 Mbps of average throughput at 17.5 µW of optical power.

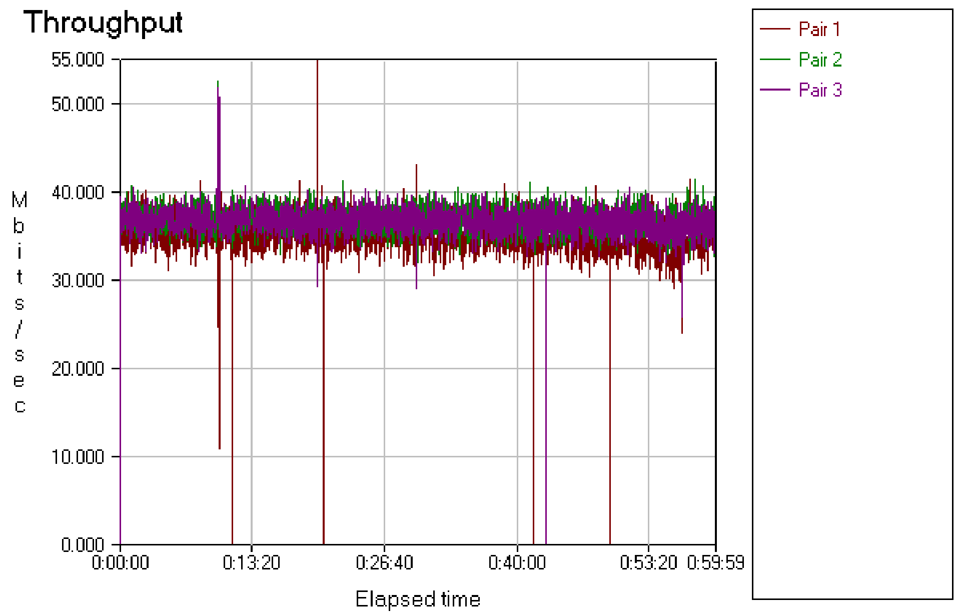
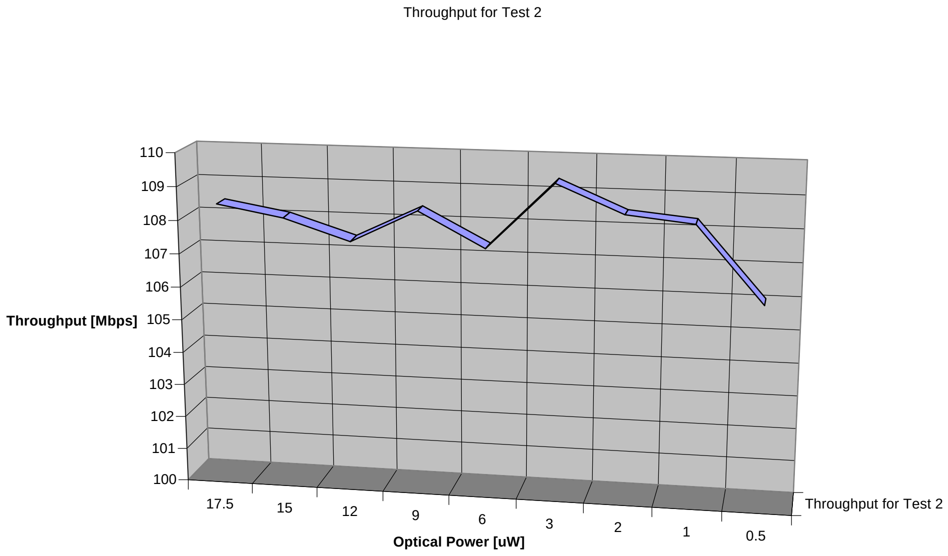


Figure 7.7 Test 2 Throughput at 17.5 mW of Optical Power

The same test run was repeated eight times at different settings of the attenuators and these values are given in Figure 7.8. From the figure, it can be observed that the average throughput also has a peculiar periodic shape over most of the operating power region.

The graph is somewhat similar to the throughput in Test 1 and its shape may also be attributed to the same two reasons mentioned for Test 1. The possibility of constructive and destructive interference of the modes, which may be dropped or reflected, may have caused oscillations in performance. Again, the overall trend of decreasing throughput, especially below the 3 μW power level has to do, as expected, with lower optical power level which decreases the signal-to-noise ratio and ultimately increases the BER.



	17.5	15	12	9	6	3	2	1	0.5
▨ Throughput for Test 2	108.412	108.063	107.439	108.412	107.374	109.366	108.513	108.313	106.046

Figure 7.8 Test 2 Average Throughput for Different Optical Power Levels

7.2.2. Test 2 Transaction Rate

A one-run trace of the Test 2 transaction rate is shown in Figure 7.9. Pair 1 had achieved around 0.44 transactions per second; pair 2 had achieved slightly above 0.46; and pair 3 had achieved slightly below 0.46 transactions per second. The overall transaction rate average for this power level was around 1.35.

Transaction rate values at different settings of the attenuators are given in Figure 7.10, from which a close resemblence to the throughput graph can be seen.

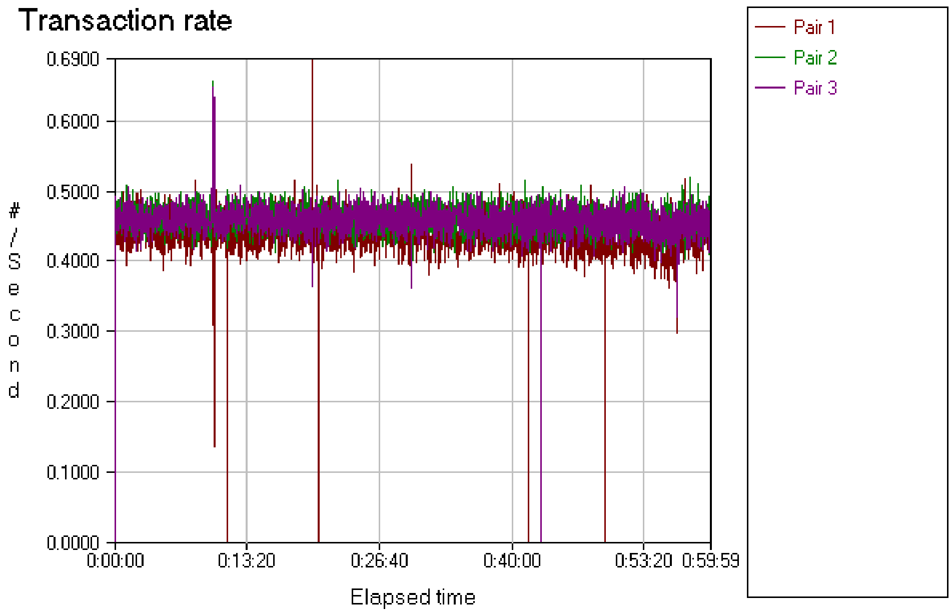
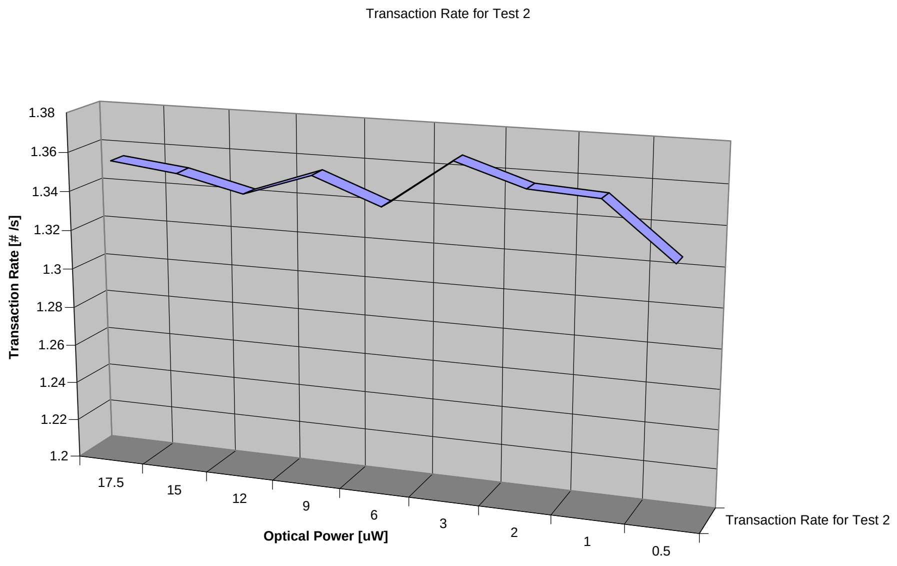


Figure 7.9 Test 2 Transaction Rate at 17.5 mW of Optical Power



	17.5	15	12	9	6	3	2	1	0.5
▨ Transaction Rate for Test 2	1.355	1.351	1.343	1.355	1.342	1.367	1.356	1.354	1.326

Figure 7.10 Test 2 Transaction Rate for Different Optical Power Levels

7.2.3. Test 2 Response Time

The response time of all three pairs is shown in Figure 7.11. Pair 1 required about 2.29 seconds per transaction, whereas pairs 2 and 3 needed about 2.18 seconds per transaction. The overall response time average for this power level was around 2.22 seconds per transaction.

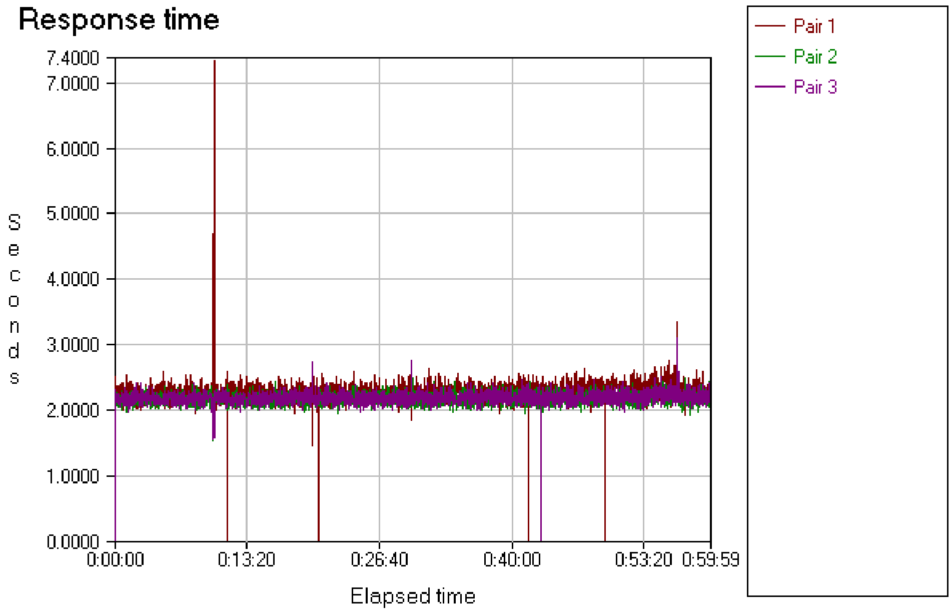
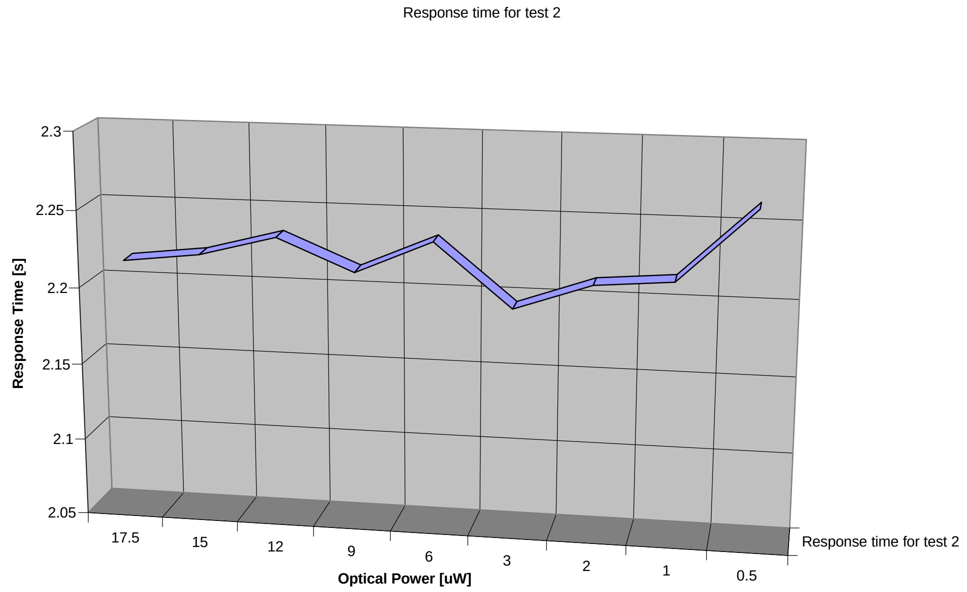


Figure 7.11 Test 2 Response Time at 17.5 mW of Optical Power

Response time values at different optical power levels are given in Figure 7.12. As was the case for Test 1, the response time graph is the opposite of the transaction rate graph. The time it takes to complete a transaction increases at lower power levels, again especially below the 3 µW of optical power level.



	17.5	15	12	9	6	3	2	1	0.5
▨ Response time for test 2	2.21518	2.22105	2.23412	2.21389	2.23544	2.19454	2.21182	2.21607	2.26332

Figure 7.12 Test 2 Response Time for Different Optical Power Levels

7.3. Test 3 Results

Test three was initially designed only to show that the soft failures may not be having as much of an effect on network performance if the link is underutilized. The overall throughput achieved with this test was much lower than any of the other three. As suspected, throughput was not affected by the link deterioration, and therefore it was unnecessary to take as many data points.

Test 3 consisted of 10 pairs running different multimedia scripts as described in Chapter 6, Section 6.4.3, and the test used UDP as the network protocol.

7.3.1. Test 3 Throughput

In Test 3, Pairs 2, 3, 4 and 5 were running scripts *netmtgvtest3, netshowutest3*, *realaudtest3*, and *realmedtest3*, respectively, and achieved throughput between 2.3 Mbps and 3.7 Mbps. Pair 1, running netmtgatest3 and Pairs 6, 7, 8, 9 and 10 all running voipstest3, achieved only around 0.3 Mbps of average throughput. These traces of individual pairs are displayed in Figure 7.13.

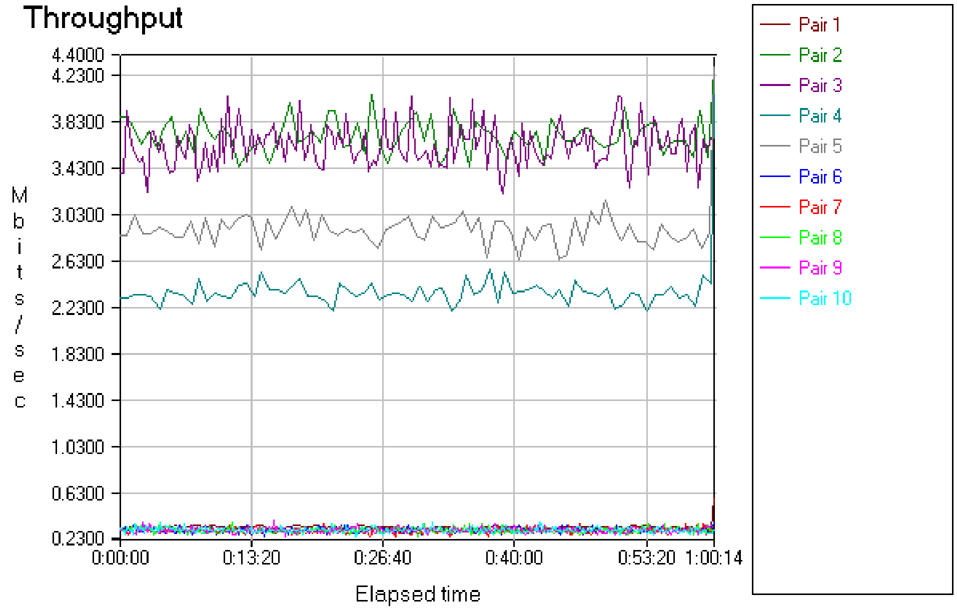
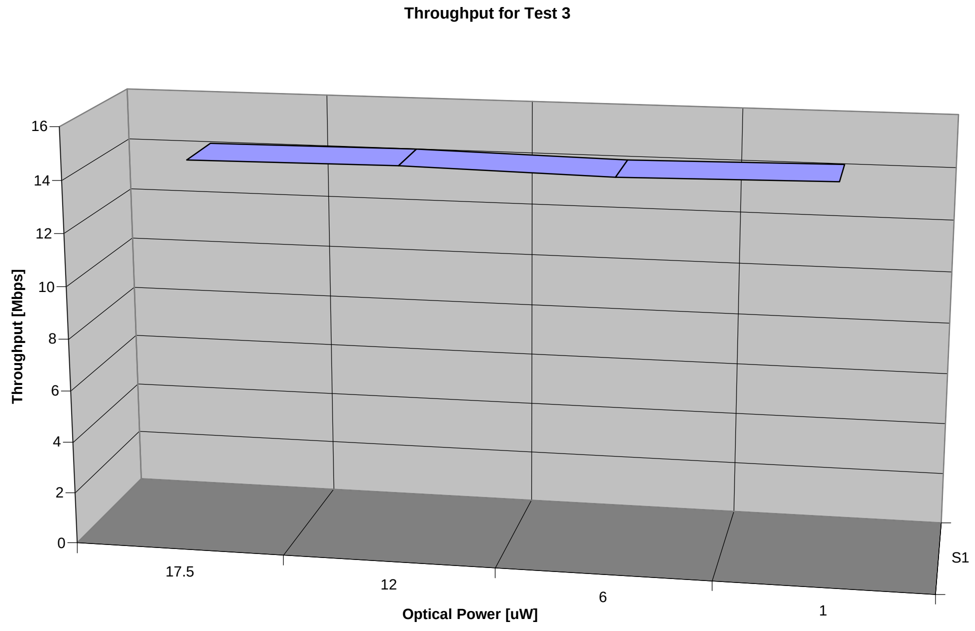


Figure 7.13 Test 3 Throughput at 17.5 mW of Optical Power

The graph of all the average values obtained at different optical power levels is presented in Figure 7.14, and as expected, does not show significant throughput change over the wide operating region of different power levels.



	17.5	12	6	1
■ Series1	14.428	14.482	14.341	14.474

Figure 7.14 Test 3 Average Throughput for Different Optical Power Levels

7.3.2. Test 3 Lost Data

The trace showing percent of data lost during the one-hour test at 17.5 μW power level for each individual pair can be found in Figure 7.15. It is not surprising that Pairs 2, 3, 4 and 5, which had achieved the highest throughput, also had the highest percentage of data lost.

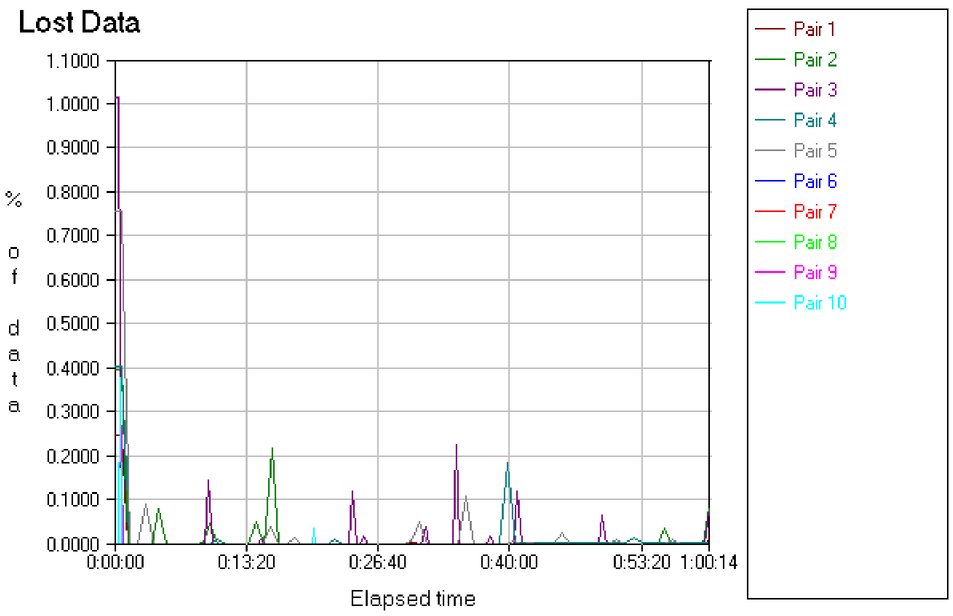
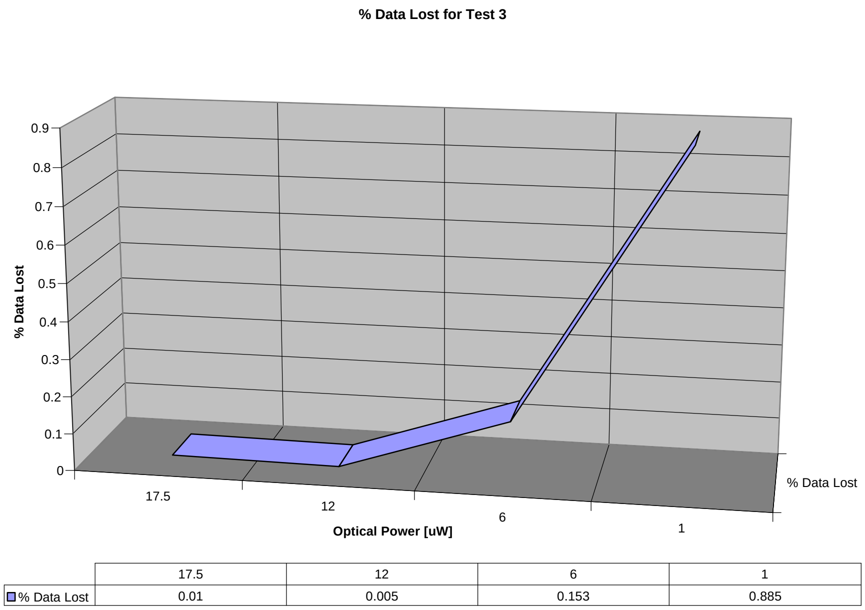


Figure 7.15 Test 3 Percentage of Lost Data at 17.5 mW of Optical Power

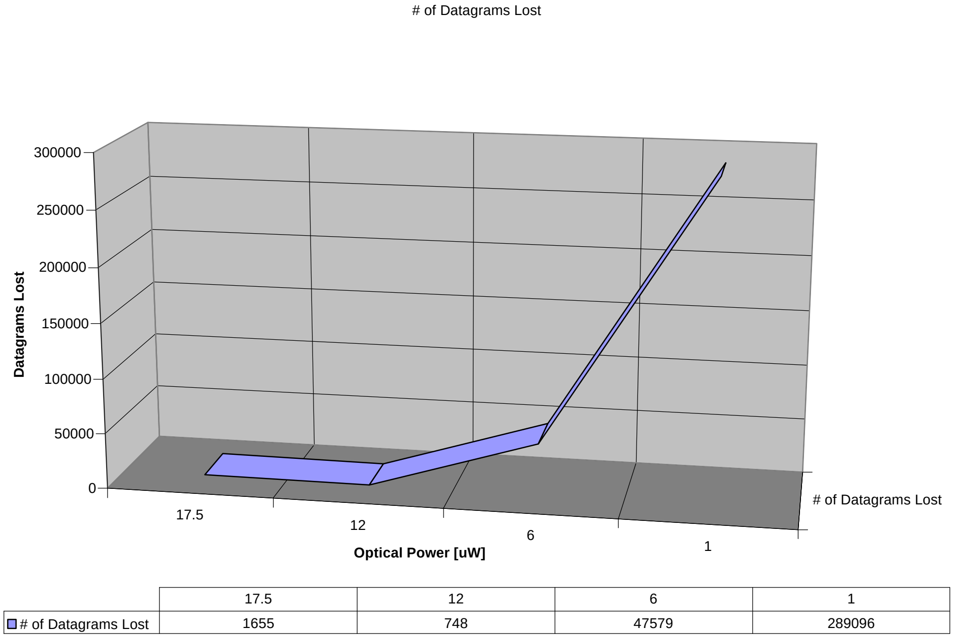
Although Test 3 was initially run only for its throughput values, which in itself showed no significant change and thus no influence of link degradation, a very interesting result came up when the lost data was analyzed and graphed: even though the throughput did not change, the lost data significantly increased as the link was deteriorated. This result shows that even though the throughput was unchanged, the amount of lost data was definitely influenced by the link deterioration.

Both, the percentage of lost data and the total number of datagrams lost over the different attenuation levels are shown in Figures 7.16 and 7.17 respectively.



	17.5	12	6	1
▨ % Data Lost	0.01	0.005	0.153	0.885

Figure 7.16 Percentage of Data Lost in Test 3 for Different Optical Power Levels



	17.5	12	6	1
▨ # of Datagrams Lost	1655	748	47579	289096

Figure 7.17 Number of Datagrams Lost in Test 3 for Different Optical Power Levels

7.4. Test 4 Results

Test 4 achieved the highest throughput of all four tests and was run over UDP. However, Test 4 was unique in that some of the runs were not complete and that congestion was so bad that the connections timed out. The selective failures made the results for this test somewhat difficult to fully represent.

7.4.1. Test 4 Throughput

Pairs 1 and 2 of Test 4 were running *iptvvstress* scripts and achieved around 36.4 Mbps of average throughput each, and Pairs 3 and 4, which were running iptvastress scripts achieved about 25.4 Mbps each, for an overall average throghput of around 123 Mbps.

Figure 7.18 shows the trace for the overall throughput of all four pairs, which were successfully completed over the one-hour test duration. However, as this test was run over the increasingly degraded link there was a point where the test could not successfully finish due to the timeout error. The likely cause for the error (CHR0245), according to Chariot, is that the partner program encountered an error while running a test, but could not report the error on the connection being used for the tests. The most common example occurs if "validate data upon receipt" is specified, which it was for this test, and then the data at Endpoint 2 does not validate correctly during the test.

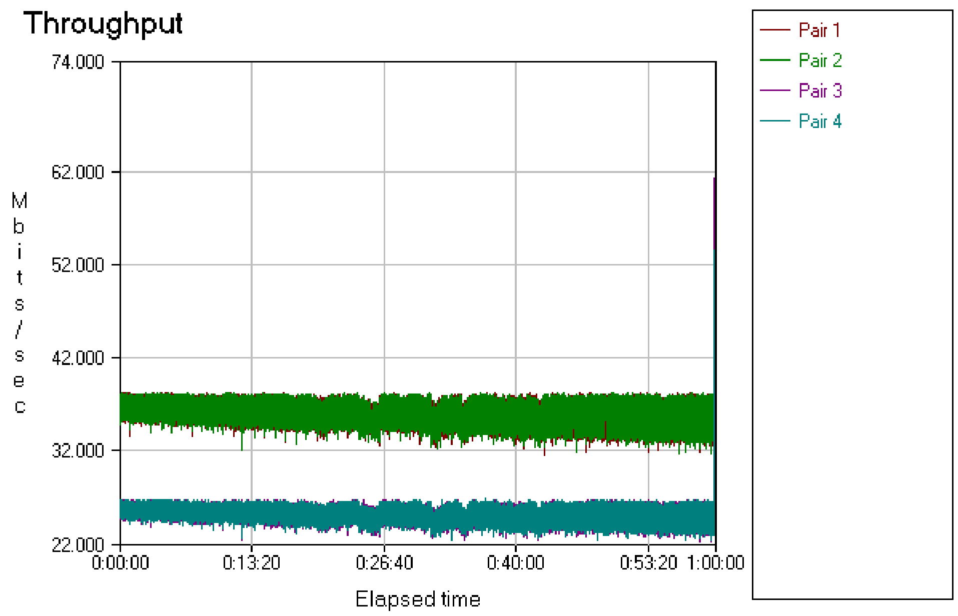


Figure 7.18 Throughput of a Successfully Completed Test 4

Essentially this error can occur when using IPX or UDP protocols. If there is a lot of traffic on the network, datagrams are lost. The endpoints will retransmit, but if things get congested enough to cause the loss of all retransmissions, the endpoints eventually time out. The first such significant error occurred at 9 µW optical power level, and the trace of the four pairs is given in Figure 7.19.

In order to ascertain the initial result the same test was repeated at the 9 µW level two more times. These traces are presented in Figures 7.20 and 7.21.

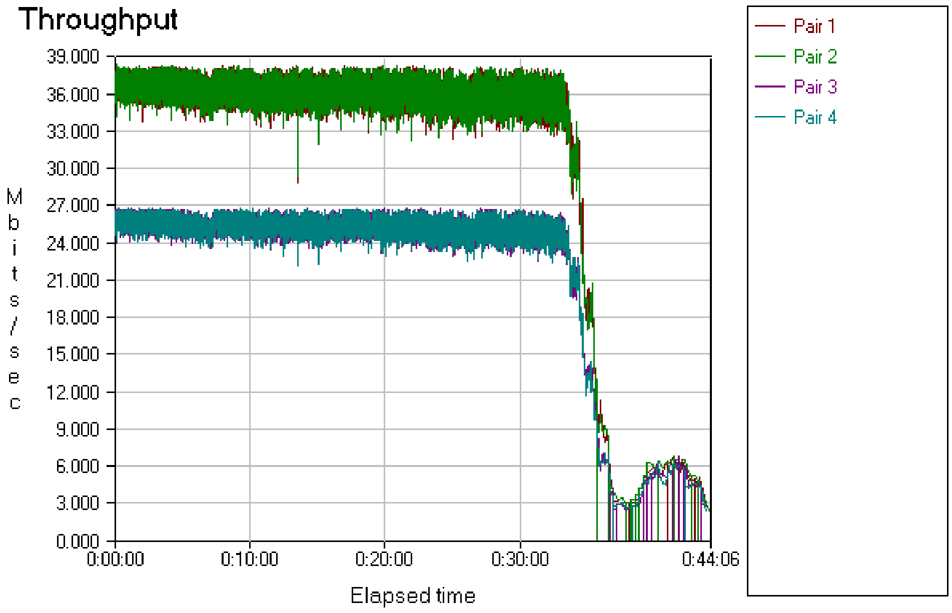


Figure 7.19 Throughput 1 of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

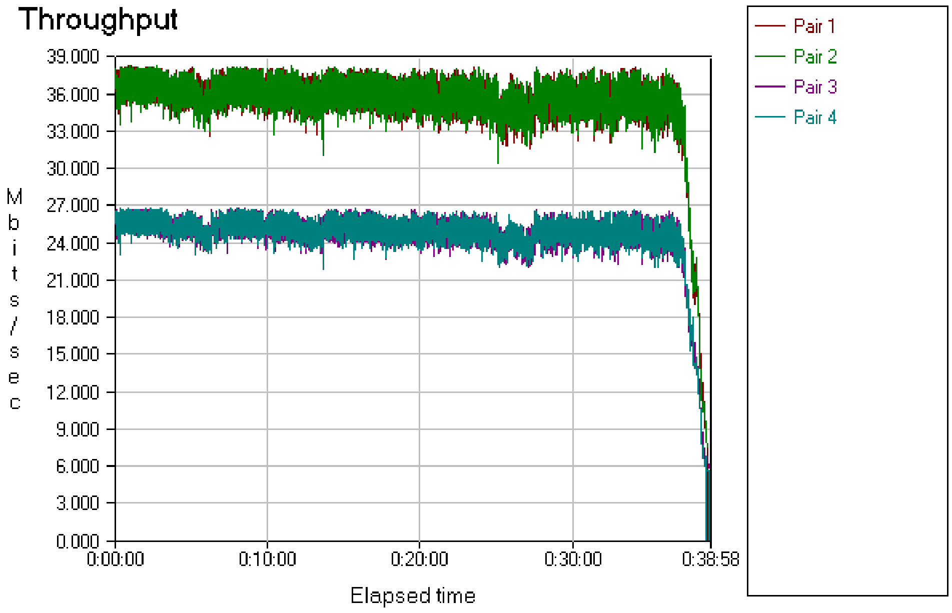


Figure 7.20 Throughput 2 of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

Although Througput 1 and Throughput 3 are similar in that both were trying to recouperate from the congestion, all three tests experienced a significant decline in throughput around the 38th minute into the test.

The graph of all the average values obtained at different optical power levels is presented in Figure 7.22. The unsuccessfully completed runs are labeled as zero throughput, whereas the successful runs display the obtained values.

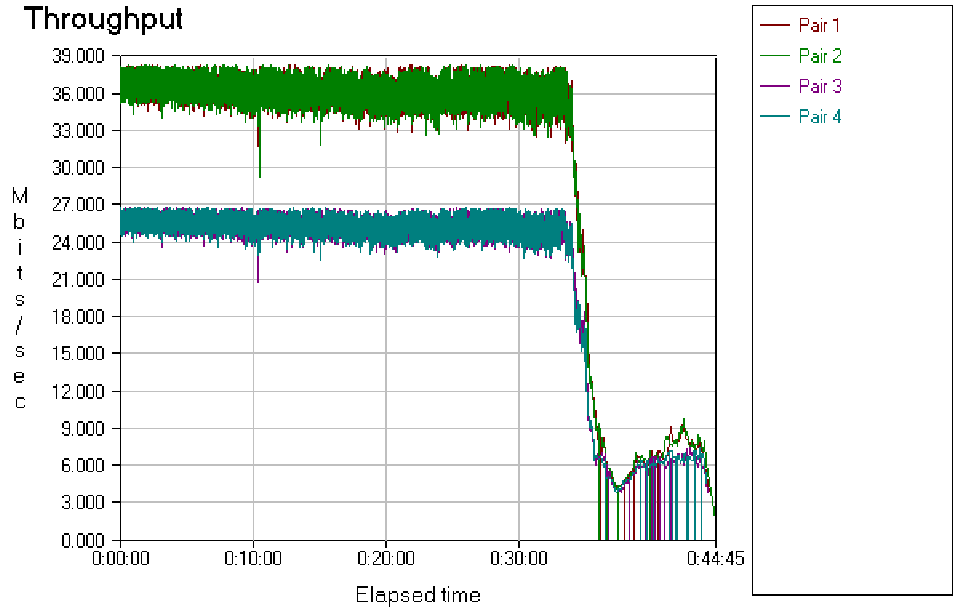
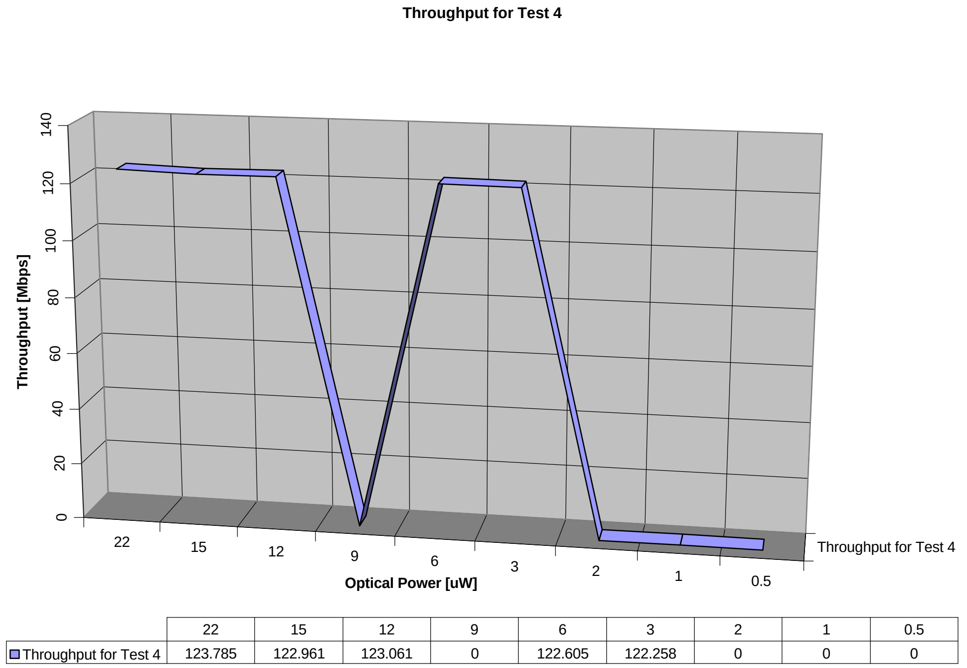


Figure 7.21 Throughput 3 of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

One of the difficulties in painting a realistic picture about this test is the fact that not all errored tests failed at the same time. Somewhat strangely, it was the test at 9 µW level that failed most quickly. The tests run at levels 2, 1 and 0.5 µW completely failed at about the 48th, 54th, and 58th minutes respectively.



	22	15	12	9	6	3	2	1	0.5
▨ Throughput for Test 4	123.785	122.961	123.061	0	122.605	122.258	0	0	0

Figure 7.22 Test 4 Average Throughput for Different Optical Power Levels

7.4.2. Test 4 Lost Data

The trace showing percent of data lost for each individual pair during the one-hour test of a successfully completed Test 4 can be found in Figure 7.23. The graph shows that the amount of lost data is somewhat uniform over the duration of the test, and it mostly stayed under 5 % of total network traffic.

A somewhat unexpected detail is that the Pairs 1 and 2, which achieved higher throughput, had a smaller percentage of data lost than the Pairs 3 and 4, which achieved lower throughput. Pairs 1 and 2 lost about 1.31 % whereas Pairs 3 and 4 lost about 1.65 % of total data.

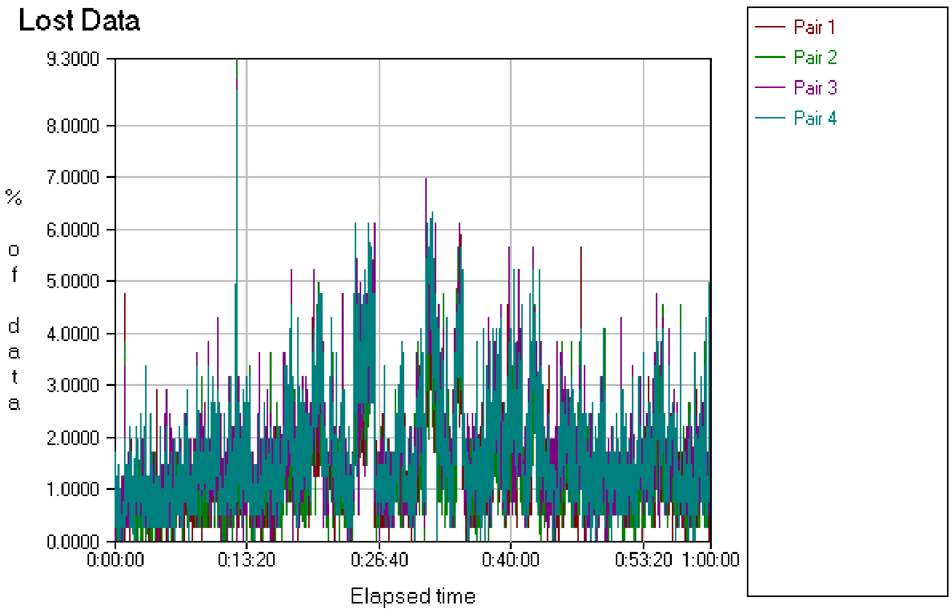


Figure 7.23 Percentage of Lost Data of a Successfully Completed Test 4

Figures 7.24, 7.25 and 7.26 show the percentage of lost data for the three failed tests, all at 9 μW of optical power. These three graphs correspond to the throughput graphs of the same unsuccessfully completed tests presented in Figures 7.19, 7.20 and 7.21.

The higher throughput Pairs 1 and 2 of the two unsuccessfully completed tests, which had the “struggling” period of trying to recover from congestion are shown in Figures 7.24 and 7.26. They lost a higher percentage of data than Pairs 1 and 2 of the tests that failed quickly or didn’t fail at all.

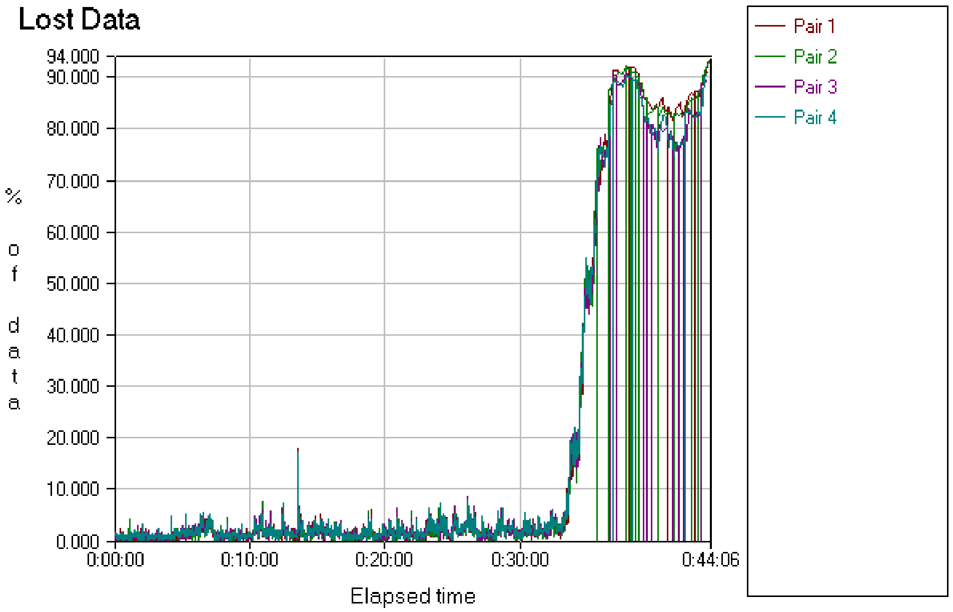


Figure 7.24 Percentage 1 of Lost Data of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

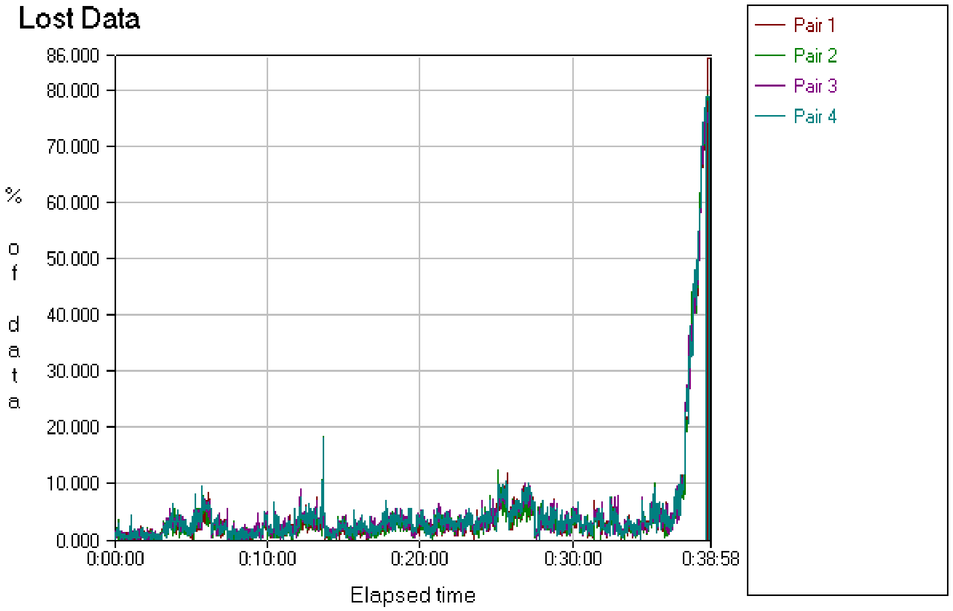


Figure 7.25 Percentage 2 of Lost Data of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

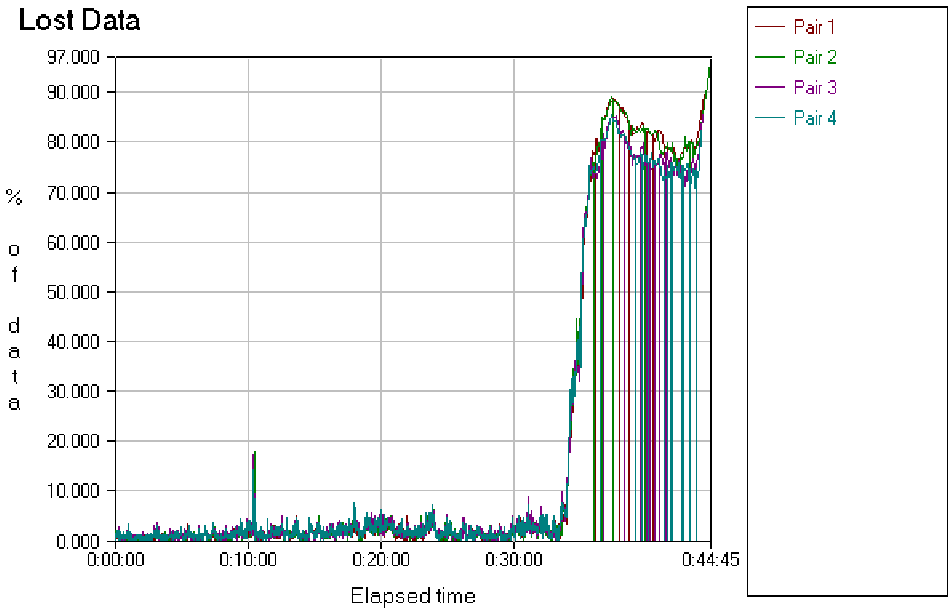
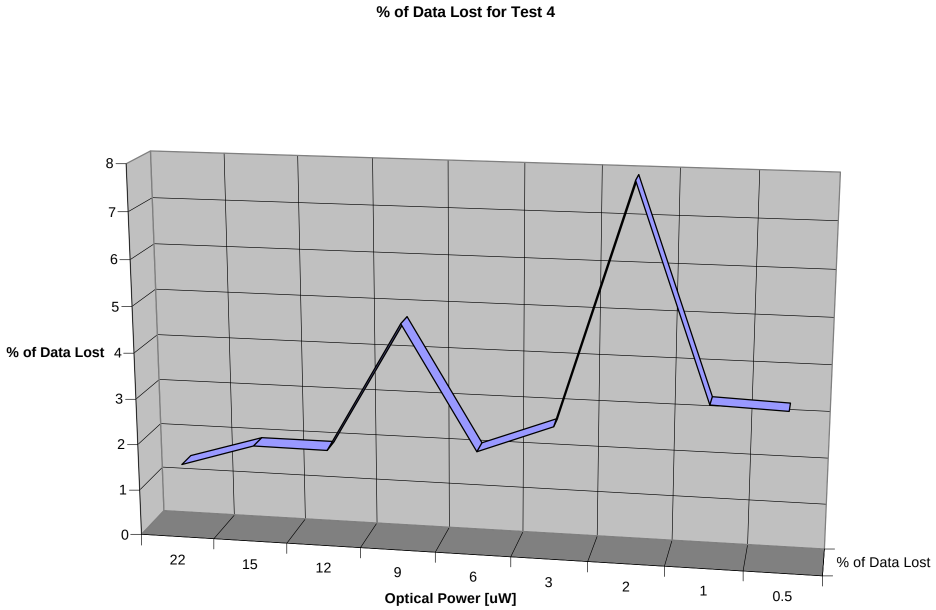


Figure 7.26 Percentage 3 of Lost Data of an Unsuccessfully Completed Test 4 at 9 mW Optical Power Level

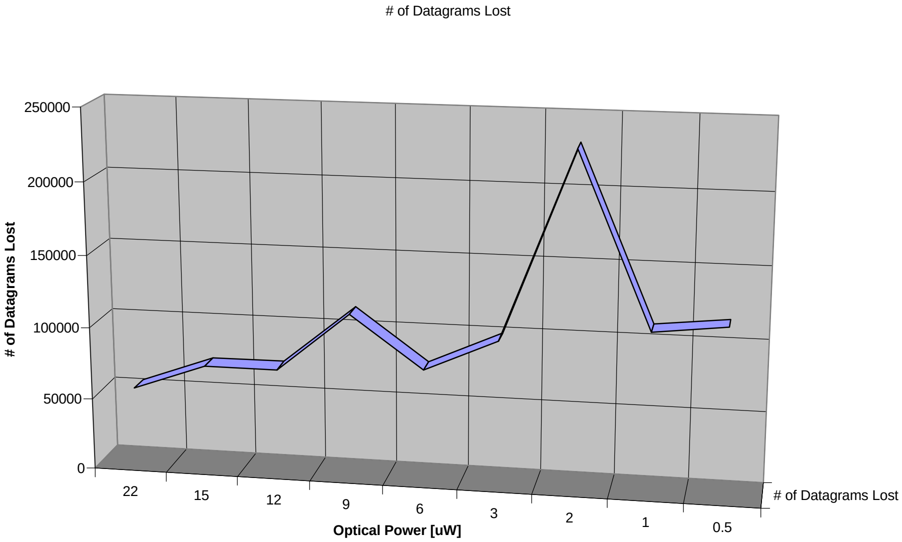
The percentage of data lost in Test 4 over the full range of optical power levels is given in Figure 7.27. The values were collected at the moment the tests either failed or finished, and represent the percent of data lost until those moments occurred.

As evident by the test at optical power level of 0.5 μW, which failed later in the test than the one at optical power level of 1 μW, it is possible to have a lower percentage of data lost even though the total number of datagrams lost is higher. This can be seen by comparing the graph of percentage of data lost shown in Figure 7.27 with the graph showing total number of lost datagrams presented in Figure 7.28.



	22	15	12	9	6	3	2	1	0.5
🟦 % of Data Lost	1.458	1.955	1.95	4.755	2.094	2.712	7.903	3.338	3.285

Figure 7.27 Percentage of Data Lost in Test 4 for Different Optical Power Levels



	22	15	12	9	6	3	2	1	0.5
🟦 # of Datagrams Lost	54161	72303	72446	113652	77369	100274	230824	111140	116881

Figure 7.28 Number of Datagrams Lost in Test 4 for Different Optical Power Levels

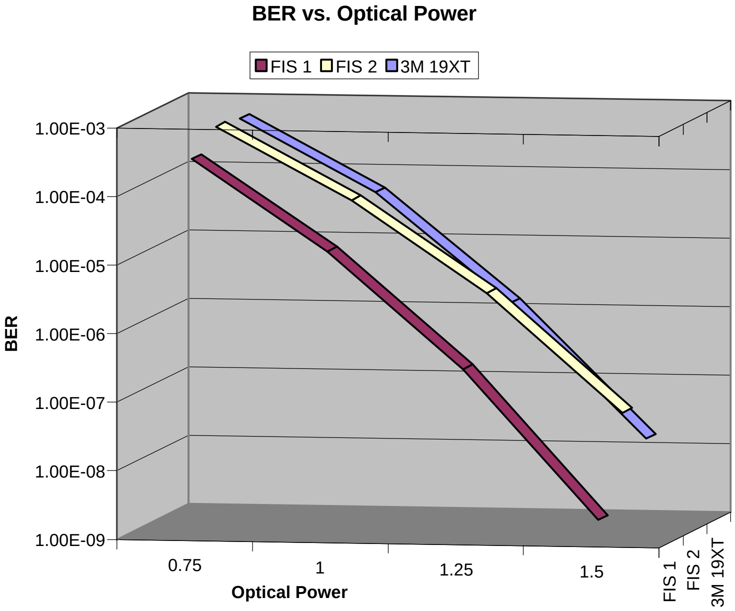
7.5. Link Characterization

Due to the observation of a nonlinear relationship between the measured performance metrics and power levels, and due to propagation peculiarities associated with short fiber links coupled with the attenuator, which acts as a modal filter, it was deemed valuable to attempt to characterize the two attenuating links used in this research project.

To achieve this characterization a transmitter with an LED at 1300 nm and corresponding receiver circuitry were assembled in the laboratory of FORCE, Inc. However, by using a different transmitter/receiver pair than the one found in the ATM switches used for this research, the obtained BER values must not be assumed representative of the entire system used for network performance testing.

The BER measurements were also done in FORCE, Inc, using their BER testing system. The bit pattern from an HP70841B Pattern Generator was fed into the transmitter, which was then connected to the link with the attenuator. After measuring the optical power coming out of the fiber, the link was connected to the receiver. The signal from the receiver was fed into an HP70842B Error Detector, which was then connected to an HP54120B Digitizing Mainframe Oscilloscope displaying the errors, as well as the eye-pattern diagram.

The signal used for this characterization was a 2^7–1 pseudo-random bit sequence over 10^{10} bits. Each run lasted 1 minute and 5 seconds, the time needed for transmitting 10^{10} bits over a link with a bit rate of 155.52 Mbps. The characterization was done on both fiber attenuating links used in this project. Since they were made by/purchased from Fiber Instrument Sales, Inc., each link is labeled in the graph as FIS1 and FIS2. To get a better representation of the way the attenuation occurs and the way the optical power is removed from a multimode fiber, the same BER tests were repeated with a 3M 19XT multimode attenuator which works by inserting a smoked mirror between the two fiber ends. The obtained results are presented in Figure 7.29.



	0.75	1	1.25	1.5
■ FIS 1	3.23E-04	1.56E-05	3.21E-07	2.18E-09
□ FIS 2	6.48E-04	5.87E-05	2.79E-06	5.36E-08
■ 3M 19XT	5.72E-04	5.18E-05	1.34E-06	1.50E-08

Figure 7.29 BER of the Link over Different Optical Power Levels

Although the BER tests were performed over the entire range of attenuation levels as used in the research project for performance testing, there were no bit errors in the links for the 65 second duration until the power level dropped to 1.5 μW. At the level below around 0.5 μW, the BER was about 0.5.

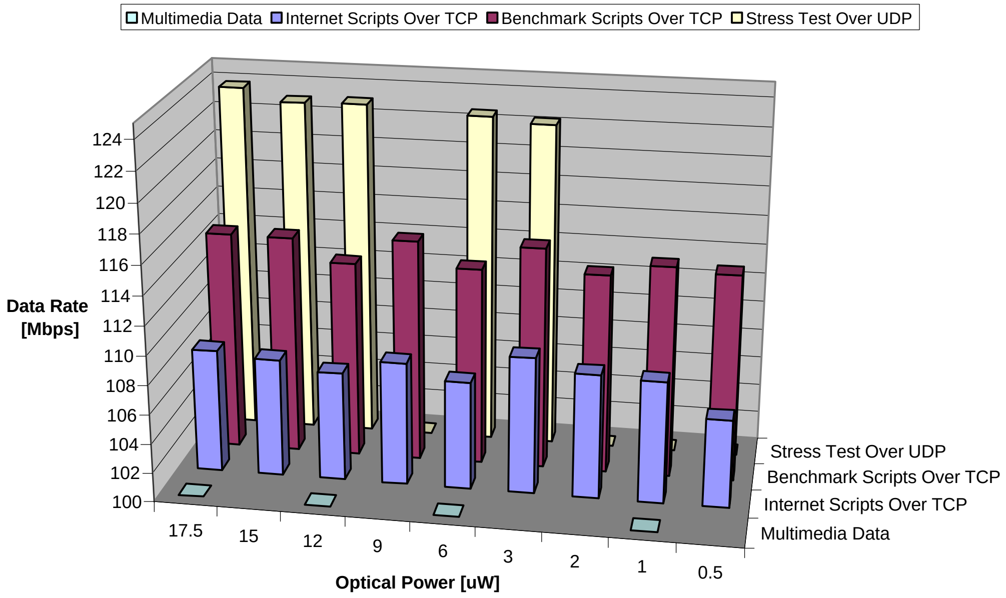
This meant that the receiver used for BER testing was not as sensitive as the receivers in the test network’s ATM switches.

Nevertheless, the graph in Figure 7.29 shows some surprising results. It can be observed that, even though the tendency of increasing the BER with decreased optical power is the same for all three links tested, they are not the same. The surprising result is that for the same optical power level two different attenuators had different BERs.

This result, coupled with the observation that the two links used for this research had significantly different BER curves, only adds to the initially stated difficulties of defining and quantizing the exact mechanisms at work in multimode fibers.

Chapter 8. Conclusions and Recommendations

A comparison graph of the throughput of the four tests is presented in Figure 8.1. Throughput was the common metric for all four tests, exibiting a similar oscilatory, yet generally decreasing trend. The unexpected performance behavior of the network shows the dependence of the higher network layers to the changes in lower network or physical layer. It is significant to note that the dependance as well as degradation in performance occur before the onset of hard failures. The dependance, no matter how complicated and nonlinear it may be, confirms the existence of “gray” area in computer network performance.



	17.5	15	12	9	6	3	2	1	0.5
□ Multimedia Data	14.428		14.482		14.341			14.474	
🟦 Internet Scripts Over TCP	108.412	108.063	107.439	108.412	107.374	109.366	108.513	108.313	106.046
🟥 Benchmark Scripts Over TCP	115.01	114.98	113.437	115.214	113.548	115.252	113.689	114.501	114.213
🟨 Stress Test Over UDP	123.785	122.961	123.061	0	122.605	122.258	0	0	0

Figure 8.1 Throughput Comparison of the Four Tests

8.1 Conclusions

Although it would be hard to find an exact relationship between the network performance and degradations in the physical layer, the results show that there is a definite influence of soft failures on end user performance. It can also be pointed out that TCP seemed to be more robust than UDP, which is not surprising based on TCP’s reliable delivery mechanisms.

It was also noticed that at the power levels at which UDP’s performance suffered the most, TCP was performing the best and vice versa. It is almost as if the UDP’s and TCP’s performance oscillations are offset by a 90° phase shift.

As witnessed from the link characterization data, the initial assumption of difficulties associated with defining and fully understanding all the phenomena interacting in multimode fibers were only further strengthened.

Horizontal cabling and fiber to the desktop dictate short fiber lengths; cheaper components and ease of use demand the use of multimode optical fibers. Coupled with an increasing demand for multimedia traffic transported over more vulnerable UDP, soft failures will likely continue to play an increasing role as yet another parameter crucial for overall network performance optimization. Therefore, the need for possible definition and classification of soft failures has been further accentuated.

8.2 Recommendations

There are several opportunities for further study. In order to better assess the influence of soft failures on network performance, it would be desirable to further load the network either by adding more endpoints and/or by adding some background traffic, such as high rate real-time video. The goal would be to assess the data transfer performance when there are multiple PVCs or SVCs servicing higher QoS traffic types.

In addition, ABR service category could be studied as opposed to the UBR used in this project. More research could determine whether the closed-loop feedback mechanism available in ABR service class would make the network performance more immune to soft failures.

From a physical link perspective, it is suspected that singlemode fibers could be more robust to soft failures such as the one introduced in this project. This is mostly due to the presence of only one propagating mode. Unfortunately the small core size makes the light sources more expensive, light coupling into the fiber more difficult and less efficient. These disadvantages may in turn make singlemode fiber more vulnerable to some other types of soft failures. Additional research analyzing network performance under soft failures of singlemode fibers may offer additional insight.

References

[1] Ali Zolfaghari, Fred J. Kaudel, “Framework for Network Survivability Performance,” *IEEE Journal on Selected Areas in Communications*, Vol.12, No. 1, pp. 46-51, January 1994.

[2] Ganymede Software, maker of Chariot network performance testing software http://www.ganymede.com/html/chariot.htm.

[3] Daniel Minoli, *Enterprise Networking*, Artech House Telecommunications Library, 1993.

[4] Michael Hauben, “History of ARPANET,” http://www.dei.isep.ipp.pt/docs/arpa.html

[5] David McDysan, Darren Spohn, *ATM Theory and Applications*, McGraw-Hill, 1999.

[6] Larry L. Peterson, Bruce S. Davie, *Computer Networks: A Systems Approach*, Morgan Kaufmann Publishers, Inc., 1996.

[7] Steve Lawrence and C. Lee Giles, “Accessibility of Information on the Web,” *Nature Magazine*, p. 107, 8 July 1999.

[8] Daniel Minoli, Anthony Alles, *LAN, ATM, and LAN Emulation Technologies*, Artech House, 1996.

[9] Rainer Händel, Manfred N. Huber, Stefan Schröder, *ATM Networks: Concepts, Protocols, Applications*, Addison-Wesley, 1998.

[10] John Henshall, Sandy Shaw, *OSI Explained: End-to-End Computer Communication Standards*, Ellis Horwood Limited, 1988.

[11] Curtis A. Siller, Jr. and Mansoor Shafi, *SONET/SDH, A Sourcebook of Synchronous Networking*, IEEE Press, 1996.

[12] ATM Forum, *LAN Emulation over ATM*, January 1995, available via anonymous ftp to: ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0021.000.pdf

[13] Stan Schatt, *Understanding ATM*, McGraw-Hill, 1996.

[14] RAD Data Communications, “ATM Tutorial,” 1994, http://www.rad.com/networks/1994/atm/tutorial.htm.

[15] William Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice-Hall, 1995.

[16] Sadie Lewis and Robert Hermes, *ATM/SONET Explained*, Numidia Press, 1995.

[17] Kai-Yeung Siu and Raj Jain, “A Brief Overview of ATM: Protocol Layers, LAN Emulation, and Traffic Management,” *Computer Communication Review*, Vol. 25, No. 2, pp. 6-20, April 1995.

[18] George Deaton, “Juggling ATM Traffic,” *Data Communications*, April 1996, http://www.data.com/tutorials/juggling_atm_traffic.html.

[19] Uyless Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995.

[20] Soung C. Liew, "A Framework for Characterizing Disaster-Based Network Survivability," *IEEE Journal on Selected Areas in Communications*, Vol. 12. No. 1, pp. 52-58, January 1994.

[21] Ryutaro Kawamura, “Architectures for ATM Network Survivability,” http://www.comsoc.org/pubs/surveys/4q98issue/kawamura.html.

[22] Tai H. Noh, “End-to-End Self-Healing SDH/ATM Networks,” *IEEE Proceedings of the 1996,* IEEE Global Telecommunications Conference, V. 3, Part 3 (of 4), London, UK, p 1877-1881, 1996.

[23] FASTAR http://www.att.com/press/0392/920325.bsa.html.

[24] Kayamakya Kyandoghere, “Survivability Performance Analysis of Rerouting Strategies in an ATM/VP DCS Survivable Mesh Network,” *Computer Communications Review*, Vol. 28 No. 5, pp. 22-49, Oct. 1998.

[25] Protection Across NEtwork Layers (PANEL) Project, http://www.uk.infowin.org/ACTS/RUS/PROJECTS/ac205.htm.

[26] Alberto Leon-Garcia, *Probability and Random Processes for Electrical Engineering, 2nd Edition*, Addison Wesley, 1994.

Glossary of Acronyms

AAL – ATM Adaptation layer
AM VSB – Amplitude Modulated Vestigial Sidebands
ANSI – American national Standards Institute
APS – Automated Protection Switching
ARPA – Advanced Research Projects Agency
ATM – Asynchronous Transfer Mode
BER – Bit Error rate
B-ISDN – Broadband Integrated Services
BT – Burst Tolerance
BUS – Broadcast/Unknown Server
CATV -- Cable Television
CBR – Constant Bit Rate
CDV – Cell Delay Variation
CI – Congestion Indication
CLP – Cell Loss Priority
CLR – Cell Loss Ratio
CP – Common Port
CPN – Customer Premises Network
CRC – Cyclic Redundancy Checksum
CS – Convergence Sublayer
CSMA/CD – Carrier Sense Multiple Access/Collision Detection
CTD – Cell Transfer Delay
EFCI – Explicit Forward Congestion Indication
EPD – Early Packet Discard
ER – Explicit rate
FDDI – Fiber Distributed Data Interface
FRVP -- Failure Resistant Virtual Path
GIF – Graphic Image Format
GFC – Generic Flow Control
HEC – Head Error Check
HIPPI – High Performance Parallel Interface
IP – Internet Protocol
ISDN – Integrated Services Digital Network
ISO – International Standards Organization
ITU –International Telecommunications Union
LAN – Local Area Network
LANE – LAN Emulation
LEC – LAN Emulation Client
LED – Light Emitting Diode
LES – LAN Emulation Server
MBS – Maximum Burst Size

MCR – Minumum Cell Rate
MPEG – Motion Picture Experts Group
MTU – Maximum Transmission Unit
NI – No-Increase
NIC – Network Interface Card
NNI – Network-to-Network Interface
nrt VBR – Non Real-Time Variable Bit rate
OAM – Operation and Maintenance
OAMP – Operations, Administration, Maintenance and Provisioning
OC – Optical Carrier
OSI – Open System Interconnection
PC – Personal Computer
PCI – Personal Computer Interface / Protocol Control Information
PCR – Peak Cell Rate
PDU – Protocol Data Units
PMD – Physical Medium Dependent
POH – Path Overhead
PPD – Partial Packet Discard
PRM – Protocol Reference Model
PT – Payload Type
QoS – Quality of Service
RAM – Random Access Memory
RM – Resource Management
RM – Resource Monitoring
rt VBR – Real-Time Variable Bit Rate
SAR – Segmentation and Reassembly
SCM – Subcarrier Multiplexing
SCR – Sustained Cell Rate
SDH – Synchronous Digital Hierarchy
SHN – Self-Healing Network
SHR – Self-Healing Ring
SMDS – Switched Multi-Megabit Data Service
SOH – Section Overhead
SONET – Synchronous Optical Network
SPE – Synchronous Payload Envelope
SS – Service Specifier
STM – Synchronous Transport Module
STS – Synchronous Transport Signal
TC – Transmission Convergence
TCP – Transmission Control Protocol
TDM – Time Division Multiplexing
TOH – Transport Overhead
UBR – Unspecified Bit Rate

UDP – User Datagram Protocol
UNI – User-to-Network Interface
UPC – Usage Parameter Control
VC – Virtual Channel
VCI – Virtual Channel Identifier
VP – Virtual Path
VPI – Virtual Path Identifier
VT – Virtual Tributary
VTxSPE – Virtual Tributary Synchronous Payload Envelope
WAN -- Wide Area Network
WDM – Wavelength Division Multiplexing

Vita Igor Aleksandar Kostic

EDUCATION

M.S. in Electrical Engineering, August 1999, Virginia Tech.
Thesis: "Fiber-Based ATM Computer Network Performance and Survivability Issues under Soft Failure Conditions."
B.S. in Electrical Engineering, Michigan Technological University.
Concentration: Communications / Electromagnetics, **Summa Cum Laude**, May 1997.
A.S. in Engineering, Bay De Noc Community College, Michigan, May 1995.

EXPERIENCE

Research Associate, Virginia Tech, January 1999-Present.
Research Engineer for Fiber & Electro-Optics Research Center (FEORC).

Job Responsibilities

- Research, analyze and present results of work on optical fiber communication systems, subsystems, and computer networks.

Projects Executed

- Completed ATM-based fiber optic network for Virginia's Smart Road, a $25+ million project:
    - Acted as liaison between the customer (primarily government agencies) and the vendors (cable, switches, etcetera).
    - Obtained equipment donations for project.
- Evaluated distance measurement system based on the wavelength shift of LD light.
- Implemented EFPI magnetic field sensors and wavelets-based signal processing for vehicle detection and classification.
- Wrote research proposals, reports, and papers.
- Gave presentations to sponsors, technical meetings, academia and industry.

Research Assistant, Virginia Tech, January 1998-December 1998.
Project Engineer for FEORC.

Job Responsibilities

- Analysis and laboratory research work of optical fiber communication systems, networks and optical fiber sensor systems.

Projects Executed

- Provided complete design specifications for the fully integrated ATM-based fiber optic network for Virginia's Smart Road.
- Delivered cable recommendations for the Smart Road.
- Investigated wavelength division multiplexing in multiwavelength networks.
- Analyzed performance of subband coding (wavelets) vs. block coding (JPEG).
- Compared fiber optic vs. coaxial cable transmission of microwave signals.

Graduate Teaching Assistant, Virginia Tech; September 1997-December 1997.

- Organized, graded, and prepared a senior level course in analog and digital communication systems.

Graduate Research Assistant, Virginia Tech, FEORC, June 1997-August 1997

- Studied analog video transmission (NTSC) -- SCM AM-VSB and digital video transmission (MPEG) -- SONET.

Other Academic Activities, Undergraduate, 1993-1997

- Interdisciplinary research.
- EE Coach.
- Math Coach.

Igor Aleksandar Kostic

TECHNICAL SKILLS

Configured ATM switches and Network Interface Cards (NIC).
Configured MJPEG codecs and routed video across private ATM networks.
Documented fiber cable layout in OrCAD.
Analyzed / tested computer network performance using Ganymede's Chariot software.
Worked with ATM Adaptation Layer multiplexers.
Performed measurements of ferrite-polymer materials as a function of frequency.
Familiarity with ATM, SONET, LANE, Ethernet, TCP / IP, UDP, FDDI.
Capabilities with Chariot, Matlab Wavelet Toolbox, OrCAD, MathCad, SPICE, Fortran.
Abilities in Windows 95 / 98, Mac OS, Linux, UNIX, SunOS, X11.

PRESENTATIONS

"***Computer Network Performance and Survivability Issues Under failure Conditions***," FEORC's 12th Annual Research Program Review, Blacksburg, VA, 27 April 1999.

"***Optical Fiber Magnetic Field Sensor and Signal Processing for Vehicle Detection and Classification***," SPIE Conference on Intelligent Systems and Advanced Manufacturing, Boston, MA, 1-5 Nov. 1998.

"***Use of Fiber Optics in Intelligent Transportation Systems***," North American Travel Monitoring Exhibition and Conference, Charlotte, NC, 11-15 May 1998.

"***Use of a Fiber Optic ATM Architecture for ITS Roadside Communications***," FHWA RCE Advisory Board Meeting, Falls Church, VA, 22-23 April 1998.

"***Fiber Optic Communication Network for the Smart Road Project and Implementation of ATM to ITS***," FEORC's 11th Annual Research Program Review, Blacksburg, VA, 16 April 1998.

HONORS

Intellectual Property Disclosure Co-Inventor, "EMI Resistant Optical Fiber Highway Barrier Breakage Sensor."
Manuscript Reviewer for Optical Society of America.
Proposal Reviewer for Virginia Center for Innovative Technology.
HKN (Eta Kappa Nu International Electrical Engineering Honor Society) Member, held office 1996 / 97.
TBP (Tau Beta Pi National Engineering Honor Society) Member.
LeaderShape Institute Graduate.
International Student Scholarship Recipient; MTU.
International Student Scholarship Recipient; Bay De Noc.
Dean's List Summer 1993-Spring 1997.

LANGUAGES

Fluent in conversational and written Serbo-Croatian and Macedonian.

ACTIVITIES

Downhill skiing, weight training.