

Research on Gigabit Gateways: Final Report

August 31, 1995

Faculty Investigators:

Victor S. Frost
Joseph B. Evans
David W. Petr
Douglas Niehaus
Gary J. Minden

Student Researchers:

Murat Bog
Cameron W. Braun
Brian Buchanan
Benjamin J. Ewy
C.N. Gupta
Marcia Ramos
R.J. Sanchez
Srini W. Seetharam
Vinai R. Sirkay
Michael T. Swink
Hugo A. Uriona
Esmaell Yousefi

TISL Technical Report TISL-9770-34

Sponsored by:

Defense Advanced Research Projects Agency/CSTO

Research on Gigabit Gateways

AARPA Order No. 8634

Issued by ESD/AVS under Contract #F19628-92-C-0080



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS N/A		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TISL-9770-34			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION University of Kansas Center for Research, Inc.		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) 2291 Irving Hill Road Lawrence, KS 66045			7b. ADDRESS (City, State, and ZIP Code) 3701 N. Fairfax Rd. Arlington, VA 22203		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Electronic Systems Division		8b. OFFICE SYMBOL (If applicable) ESD	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-92-C-0080		
8c. ADDRESS (City, State, and ZIP Code) Air Force Systems Command, USAF Hanscom AFB, MA 01731-5000			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Report on Gigabit Gateways: Final Report (Unclassified)					
12. PERSONAL AUTHOR(S) Victor S. Frost, Joseph B. Evans, David W. Petr, Douglas Niehaus, Gary J. Minden					
13a. TYPE OF REPORT Final Tech Report		13b. TIME COVERED FROM 6/10/92 TO 8/31/95		14. DATE OF REPORT (Year, Month, Day) 95,8,31	
15. PAGE COUNT 113					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The use of similar technology in local and wide area networks enables geographically distributed high performance applications. Key elements in achieving high performance are the appropriate use of traffic control and the development of efficient gateways between LANs and WANs. Even though the basic technology used on both sides of a gateway may be similar, the operational aspects of these elements are significantly different. The performance of the MAGIC WAN was evaluated, and bottlenecks were identified and analyzed. Techniques were developed and implemented, specifically ATM cell level pacing, to eliminate these bottlenecks. Throughput performance close to the theoretical maximum was demonstrated. The MAGIC network enabled the development of a software system that allows the remote viewing and control of standard XWindows-based applications; its performance is reported here. A gateway has also been developed and implemented to not only support communications between and ATM LAN and WAN at 622 Mb/s but to provide a platform for conducting continued					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Victor S. Frost			22b. TELEPHONE (Include Area Code) 913/864-4833		22c. OFFICE SYMBOL

19. continued

network control and traffic research. This report will describe experiences with ATM over a WAN and how the gateway was developed, implemented and evaluated. The results included here will show how high speed LAN/WAN internetworking can be achieved and applied in many environments as appropriate control techniques and interfaces become ubiquitous.

Table of Contents

	Page
List of Figures	ii
List of Tables	iii
Abstract	iv
1.0 Executive Summary	1
2.0 Project Overview	2
3.0 Technology Transfer	4
4.0 A 622 Mb/s SONET/ATM-LAN/WAN Gateway	5
4.1 Gateway Overview	5
4.2 Gateway Architecture	6
4.3 Gateway Results	11
5.0 Real-Time Traffic Control in ATM WANs	15
5.1 The Need for ATM-Layer Traffic Control in WANs	15
5.2 Static ATM-Layer Traffic Control for ATM WANs	15
5.3 Development, Design, and Implementation of Real-Time Traffic Control for ATM WANs	16
6.0 The Over-The-Shoulder Application	22
6.1 Architecture of the OTS Application	23
6.2 Performance of the OTS Application	24
6.3 OTS: Lessons Learned	27
7.0 A Distributed Network Emulation System	31
8.0 An Overview of Tools to Support Gigabit Network Performance Evaluation	33
9.0 Lessons Learned	34
10.0 References	36
Appendix A	37
Appendix B	38
Appendix C	39
Appendix D	40
Appendix E: Project Publications	41



List of Figures

	Page
Figure 4.1 AN2/SONET Gateway	6
Figure 4.2 Block Diagram of Transmit Section of Gateway	7
Figure 4.3 Block Diagram of Receive Section of Gateway	8
Figure 4.4 Block Diagram of ATM Support Chips in Gateway	9
Figure 4.5 Histogram of Cell Interarrivals for Unpaced Stream	12
Figure 4.6 Histogram of Cell Interarrivals for Paced Stream	12
Figure 4.7 Time Series of Cell Arrivals for Paced and Unpaced Streams	13
Figure 4.8 OC-12c WAN Experiment	13
Figure 5.1 OTTO Pacing Scheme	16
Figure 5.2 Measurement Points on the ATM/SONET Gateway Card	17
Figure 5.3 Adaptive i-out-of-m Shaping Algorithm	19
Figure 5.4 Rate Adaptation of Shaper to Ethernet Trace Data	21
Figure 6.1 Client/Server Architecture for the OTS Application	23
Figure 6.2 Configuration of OTS Experiments	25
Figure 6.3 Protocol Architecture	25
Figure 6.4 Frame Traffic Without Congestion	28
Figure 6.5 Frame Traffic With Congestion	28
Figure 6.6 Histogram of Frame Interarrival Times Without Congestion	29
Figure 6.7 OTS: Histogram of Frame Interarrival Times With Congestion	29
Figure 6.8 OTS: Histogram of AAL5 PDU Interarrival Times Without Congestion	30
Figure 6.9 OTS: Histogram of AAL5 PDU Interarrival Times With Congestion	30



List of Tables

	Page
Table 4.1 Summary of the Features of the ATM/SONET Gateway	11
Table 4.2 Gateway Throughput	14
Table 6.1 Effects of Cross Traffic on the OTS Application	26



Abstract

The use of similar technology in local and wide area networks enables geographically distributed high performance applications. Key elements in achieving high performance are the appropriate use of traffic control and the development of efficient gateways between LANs and WANs. Even though the basic technology used on both sides of a gateway may be similar, the operational aspects of these elements are significantly different. The performance of the MAGIC WAN was evaluated, and bottlenecks were identified and analyzed. Techniques were developed and implemented, specifically ATM cell level pacing, to eliminate these bottlenecks. Throughput performance close to the theoretical maximum was demonstrated. The MAGIC network enabled the development of a software system that allows the remote viewing and control of standard XWindows-based applications; its performance is reported here. A gateway has also been developed and implemented to not only support communications between an ATM LAN and WAN at 622 Mb/s but to provide a platform for conducting network control and traffic research. This report will describe experiences with ATM over a WAN and how the gateway was developed, implemented and evaluated. The results included here will show how high speed LAN/WAN internetworking can be achieved and applied in many environments as appropriate control techniques and interfaces become ubiquitous.



1.0 Executive Summary

The Research on Gigabit Gateways project was a three-year effort conducted as part of the Multidimensional Applications and Gigabit Internetwork Consortium (MAGIC) effort. The goals for MAGIC were to develop a very high-speed, wide-area networking testbed to address the challenges of gigabit wide-area networking and demonstrate real-time, interactive exchange of data at gigabit-per-second (Gb/s) rates among multiple distributed servers and clients (see [1-3] and <http://www.magic.net/> for details about MAGIC). The MAGIC research project established a standards-based, gigabit-per-second internetworking testbed that supports research and development for next generation networking technologies and applications, demonstrated the feasibility of utilizing a gigabit-per-second wide-area network (WAN) to implement a defense-related application that requires massive amounts of remotely stored and processed data in real-time (the TerraVision application [4] and the supporting Image Server System [5]), and explored and resolved some of the technical challenges and issues associated with creation and use of such a high-speed, distributed computing environment. The Research on Gigabit Gateways project was responsible for the development of a gateway between the MAGIC WAN and a Digital Equipment Corp. (DEC) AN2 local-area network (LAN), development of associated resource allocation and dynamic control mechanisms, the characterization of the network through network performance and traffic measurements, and the demonstration of the feasibility of multi-location viewing of the screens generated by a user of the terrain visualization application.

The MAGIC WAN to DEC AN2 LAN OC-12c (622Mb/s) gateway was successfully developed and tested. The gateway includes measurement and resource allocation and adaptive control mechanisms. These adaptive algorithms indicated that it is feasible to dynamically control asynchronous transfer mode (ATM) flows at high rates (622 Mb/s).

The performance of the MAGIC WAN was evaluated, and bottlenecks were identified and analyzed. Techniques were developed and implemented, specifically ATM cell level pacing, to eliminate the bottlenecks. Throughput performance close to the theoretical maximum was demonstrated.

The throughput demonstrated as part of this effort clearly showed that high performance ATM LAN/WAN internetworks are feasible.

To support the network analysis activities, new ATM measurement tools were developed.

A multi-location viewing of the screens generated by a user of the terrain visualization application was developed and demonstrated.

2.0 Project Overview

Networks based on asynchronous transfer mode are evolving rapidly. In 1992, when MAGIC was formed, there were few ATM products and no ATM wide-area network (WAN) services. In the span of three years many products, host interfaces, and switches have been introduced, and ATM WAN services are now available. The initial host interfaces and corresponding switches operate at DS3 (45 Mb/s), TAXI (100 Mb/s) or OC-3c (155 Mb/s) line rates. Existing ATM WAN service is at DS3 (45 Mb/s). In parallel to this rapid development, research results were produced as part of this effort that have advanced the evolution of ATM technology.

The objective of this research was to create a testbed to internetwork diverse gigabit local-area networks (LANs) and a wide-area network (WAN) for the development and implementation of resource allocation and control mechanisms and to quantitatively characterize the performance and traffic on an LAN/WAN internetwork. The testbed also provided the framework for the development of a terrain visualization application, TerraVision, that was created by SRI and is described in [4]. TerraVision uses a distributed image server systems that was developed by Lawrence Berkeley National Laboratory [5]. As part of this effort, a distributed multi-person/multi-location interactive terrain visualization application was developed.

TerraVision (a real-time terrain visualization system) has been the forcing application for MAGIC. This system permits a commander to "drive through" a battlefield, potentially increasing effectiveness by improving the ability to "see" the battlefield and to share this common view of the battlefield with others. To facilitate such sharing, an innovative "over-the-shoulder" view of the real-time terrain visualization has been developed and demonstrated as part of this effort. Network rates on the order of 300 Mb/s are needed to support the advanced features of TerraVision. These rates presented a challenge when the project started, and even with the rapid development of ATM technology are not commonly available today.

To demonstrate that these rates can be obtained in an ATM LAN/WAN internetwork, KU has designed, implemented, and tested a flexible OC-12c (622 Mb/s) gateway between a gigabit LAN and WAN. The target LAN was the AN2, an ATM switch provided by Digital Equipment Corp., while the WAN is the MAGIC ATM-SONET (Synchronous Optical NETwork) network. Two AN2 switches are operational at KU. These have been connected to the MAGIC network using both OC-12c and OC-3c interfaces. KU used field programmable gate arrays (FPGA) in the gateway design. When combined with the on-board R3000 line card processor, the FPGAs allowed KU to implement LAN/WAN resource allocation and control mechanisms as well as the collection of performance statistics, e.g., ATM cell flows. The gateway interfaces to the WAN through either a single OC-12c or four STS-3c streams multiplexed on one OC-12.

Bottlenecks in TCP/IP over ATM WANs were identified in early experiments with TerraVision. These experiments used multiple OC-3c links. As part of this research, the cause of this congestion was identified and traffic shaping (pacing) algorithms were

developed, implemented, and demonstrated to allow full use of available link capacity. A set of measurement tools was developed and applied to quantify the performance of the MAGIC network and determine the use of available capacity in the presence of congestion.

ATM offers the potential to provide customers bandwidth-on-demand and other dynamic services over LANs and WANs. It is the first technology with the prospect to allow seamless migration from the local to the wide area. To reach its potential there must be clear demonstrations of impressive performance across LAN/WAN boundaries. This project clearly proved that the information transport capabilities of ATM were preserved in moving from an ATM LAN to a WAN. Further, the success of the OC-12c gateway showed that these capabilities will scale to high speeds.

The following chapters will provide an overview of results on: technology transfer, the OC-12c gateway, traffic pacing and real-time traffic control at 622 Mb/s, the "over-the-shoulder" application, a distributed emulation system for rapid development of network hardware and software, and tools for the quantitative evaluation of ATM systems. Details on these topics are provided in papers included in Appendixes A-D. A complete list of technical reports, papers, and theses produced as part of this effort is included in Appendix E. Additional information about MAGIC can be found at <http://www.magic.net/KU.html>.

3.0 Technology Transfer

The traffic pacing techniques developed and tested in MAGIC enabled the ACTS ATM Internetwork (AAI) to achieve near-theoretical TCP throughput.

The tools to quantify ATM traffic, specifically a tool to measure ATM traffic at the host interfaces, have been used by Digital in the AAI system and by other ATM researchers.

The OC-12c gateway has been used at Sprint Corp., and it is anticipated that the OC-12c gateway will be used in other networks.

A network emulator called DISNES (Distributed Network Emulation System) was designed and developed. DISNES was based on previous software developed at Digital Equipment Corporation's Systems Research Center (SRC) and was customized for the modeling of the AN2 network. Later, SRC modified and adapted DISNES to meet the evolving needs of the AN2 network model. DISNES was used by SRC to validate the overall design and analyze the effect of the FlowMaster flow control mechanism, iterative matching, and smooth scheduling algorithms. The simulator was then available as a tool for other parties inside Digital with a desire to further explore the performance of the AN2 network. In particular, Digital's Network Engineering (Littleton, Mass.) expressed interest in using the tool and possibly adapting it to model future versions of the AN2 network.

A graduate course on the implementation of high performance integrated networks was taught to about 50 students. The course covered processing requirements for integrated networks and associated applications; principles of VLSI architectures; an overview of selected network functions, including scrambling and descrambling, synchronization, cell switching, routing, bandwidth shaping and policing, encryption, and decryption; and implementation of network functions using high-performance special-purpose architectures. This course included projects like FPGA designs for ATM network components and IP multicast software.

The MAGIC network was used to deliver a senior/first semester graduate level course on communications networks from the KU campus in Lawrence to Sprint Corp. in Overland Park, Kansas. The course information, text, graphics, audio, and video were delivered using standard Internet technology, IP over ATM. The complete course notes were provided on the World Wide Web. At the Sprint facility the notes obtained from the Web were displayed on two screens with the video from KU on a third.

Portions of the OTTO device driver pacing code developed at KU, in particular the code that generated CBR schedules for specified bandwidth settings, was used at other sites as part of the DEC ASAP program.

4.0 A 622 Mb/s SONET/ATM-LAN/WAN Gateway

The use of similar technology in the LAN and WAN environments provides the opportunity for geographically distributed high-performance networks. A key element in realizing this goal is the development of efficient gateways, or user-network interfaces (UNIs), between the LAN and WAN environments; although the basic technology used on both sides of the gateway may be similar, the operational aspects of LANs and WANs are significantly different. A gateway that accommodates these differences has been designed and implemented. The purpose of this section is to explain the features and capabilities of the gateway as well as provide a high-level view of its implementation. Appendix A contains a description of the OC-12c gateway.

4.1 Gateway Overview

The target gigabit LAN in this work was the AN2 provided by Digital Equipment Corporation and developed by the DEC Systems Research Center [6]. The AN2 is a local area network based on ATM technology. The LAN network consists of switches (two in out network) and DECStation 5000 and DEC Alpha hosts equipped with AN2 OC-3c adapter boards (OTTOs). These hosts communicate locally via the AN2 switches and with remote MAGIC sites via the LAN/WAN gateway.

The gateway supports B-ISDN ATM traffic between the KU local-area network and the MAGIC wide-area network at SONET OC-12 or OC-12c rates (622.08 Mb/s). The AN2/SONET gateway supports:

- OC-12c interface to the wide area network;
- 4 x OC-3c streams contained in an OC-12 interface to the wide area network;
- operation within the DEC AN2 local ATM switch by connecting to the AN2 switch backplane;
- implementation of both rate- and credit-based WAN flow control techniques;
- experimental techniques for dynamic bandwidth allocation;
- experimental techniques for interoperability between connection-oriented and connectionless protocols; in particular, the gateway supports TCP/IP traffic, but it is not restricted to that protocol suite; and
- measurement of network performance.

The AN2/SONET gateway is a single card that plugs into an AN2 switch port. The gateway, shown in Figure 4.1, contains three primary subsystems: the receive section, the transmit section, and the line card processor (LCP) section.

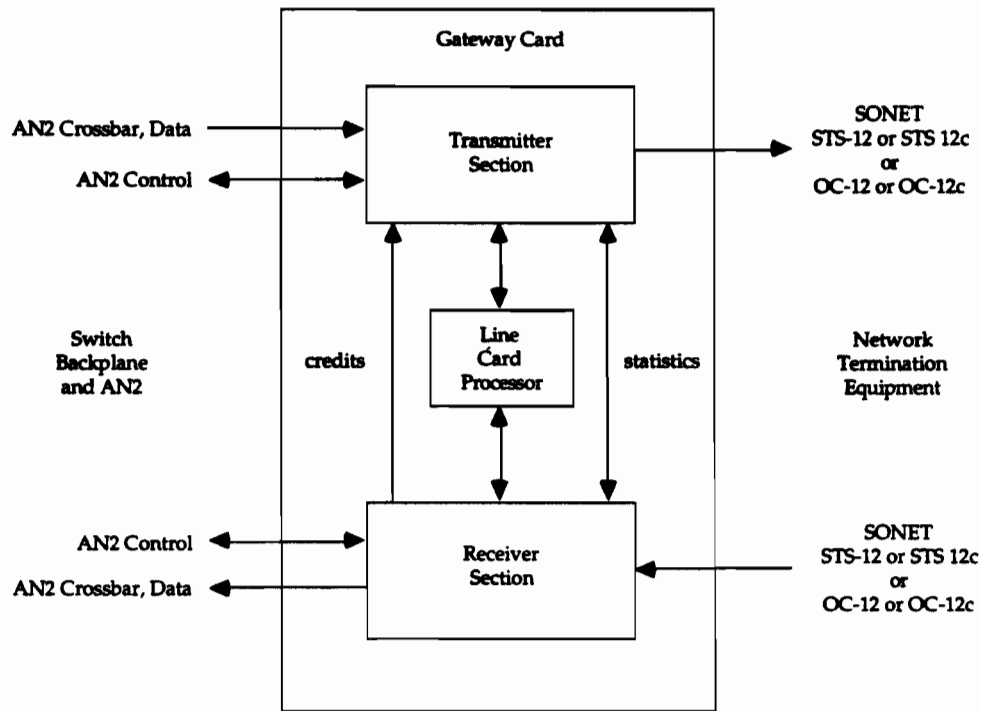


Figure 4.1 AN2/SONET Gateway

4.2 Gateway Architecture

A block diagram of the transmit section of the gateway is shown in Figure 4.2. As the cells come off the AN2 crossbar, rate adjustment is performed from the 800 Mb/s AN2 LAN speed to the 622 Mb/s SONET WAN. The data interface between the AN2 and the gateway is 32 bits wide to keep the clock frequency low in both the OC-12c and quad OC-3c designs. The clock period on the AN2 is 40 ns, while the SONET data operates at the 51.2 ns clock speed for 32-bit data. This, along with the fact that we supported four SONET OC-3c streams, necessitated that we use FIFO buffers. Specifically, four FIFO buffers are used, one for each stream in the quad OC-3c mode. In the OC-12c mode, the FIFOs are filled and serviced in a round-robin mechanism on a per-cell basis. The writing of the FIFOs is controlled by a field programmable gate array (FPGA) in the transmit path, aptly named the FIFO controller. This chip resets the FIFOs, initializes them under the command of the line card processor (LCP), and then proceeds to fill the FIFOs with cells received from the AN2. It also contains a free-running counter that is used to generate signals to synchronize the overhead and payload sections of the SONET frame. In general, these signals provide an indication of the start of the frame, the start of the row within a frame, and the location of overhead within the SONET frame.

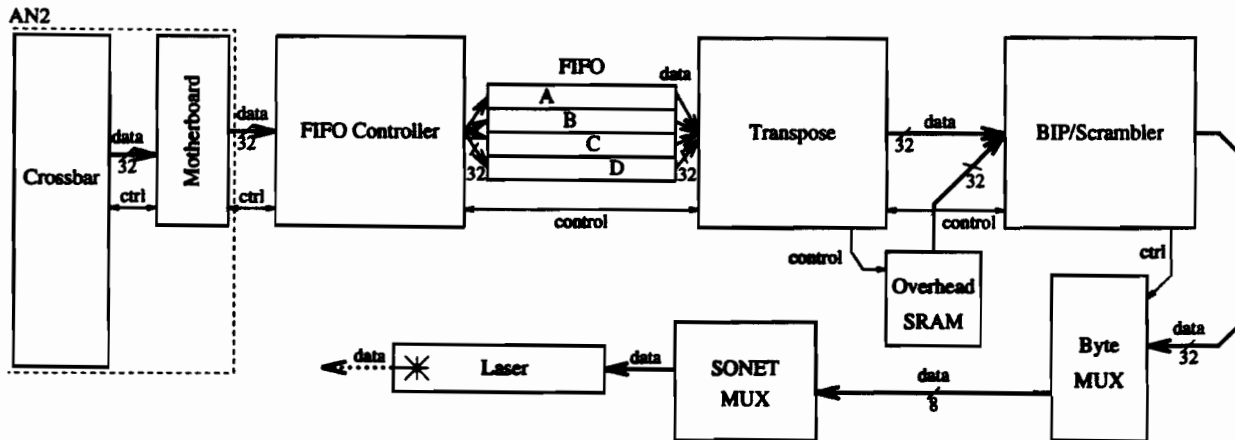


Figure 4.2 Block Diagram of Transmit Section of Gateway

The transpose is the second FPGA in the transmit path. The transpose is named for its primary function in the quadruple OC-3c mode. It performs the byte multiplexing of the four OC-3c streams into an OC-12 stream. This operation is carried out by reading a single word (32 bits each) from each of the FIFOs and then performing a transpose operation on them. These words originally came from four different cells; hence the byte-wide multiplexing effect. The transpose inserts idle ATM cells into the streams when the FIFO is empty, and it performs ATM-level cell synchronous scrambling, computation, and insertion of the header error correction (HEC) byte and the insertion of the section, line, and path overhead into the SONET frame. The transpose implements these functionally within a pipelined architecture. This implies that the signals that synchronize the SONET frame need to be delayed by the pipeline depth before they are passed to the third and final FPGA of the transmit path.

Most of the SONET section, line, and path overhead is stored in dual ported SRAMs. This enables the LCP to write in the fields that do not require high-speed processing in order to be computed. In order to simplify this entire process, pointers within the line overhead are always known and never change. The LCP can write these once and then never change them. The fields in the overhead SRAM are arranged to simplify their insertion into the datapath. With 12 bits of address, the least significant four bits represent the column within the SONET overhead, the middle nibble represents the row of the frame, and the high nibble represents the frame number. This meant that most of the overhead for 16 frames could be stored in the SRAM, with only a few fields requiring updates at relatively low service rates.

The BIP/Scrambler performs the most intensive computation on the SONET frame structure. This is the bit interleaved parity (BIP) for the section, line, and path regions of the frame. Each of these fields is a parity on all the bits of that particular region taken in either a byte-wide or integer number of byte-wide chunks. The section and path BIP are one byte per SONET payload, hence they are one byte wide in the OC-12c mode, while they are four bytes wide in the quad OC-3c mode. The line BIP is a total of 96 bits wide in both cases. The second and most important function of the BIP/Scrambler is the

application of the SONET PN sequence to scramble the bits before they are transmitted over the fiber-optic medium. This is performed in parallel fashion 32 bits at a time. Finally the BIP/Scrambler controls an external counter running at four times the word clock (51.2 ns) rate. This counter runs off the SONET byte clock (12.86 ns) and controls a bank of multiplexers that convert the data path from 32 bits to 8 bits. The 8-bit data is fed into a SONET mux, through an AT&T 1227 laser, and then into the fiber-optic link.

The receiver structure is shown in Figure 4.3. The SONET byte stream is fed into several registers which converts the byte stream to a 32-bit word stream. The word stream is then forwarded to the descrambler FPGA. This chip extracts the section BIP, descrambles the input stream by applying the SONET PN (pseudo noise) sequence, and computes and compares the section BIP. The syndrome is substituted for the BIP value. The descrambler also provides an interface to the LCP for it to monitor the status of SONET framing information and other hardware interrupts that may be generated. The descrambler generates the SONET synchronizing signals as well as the most significant eight bits (out of 12) of address required for the receive SONET overhead SRAMs.

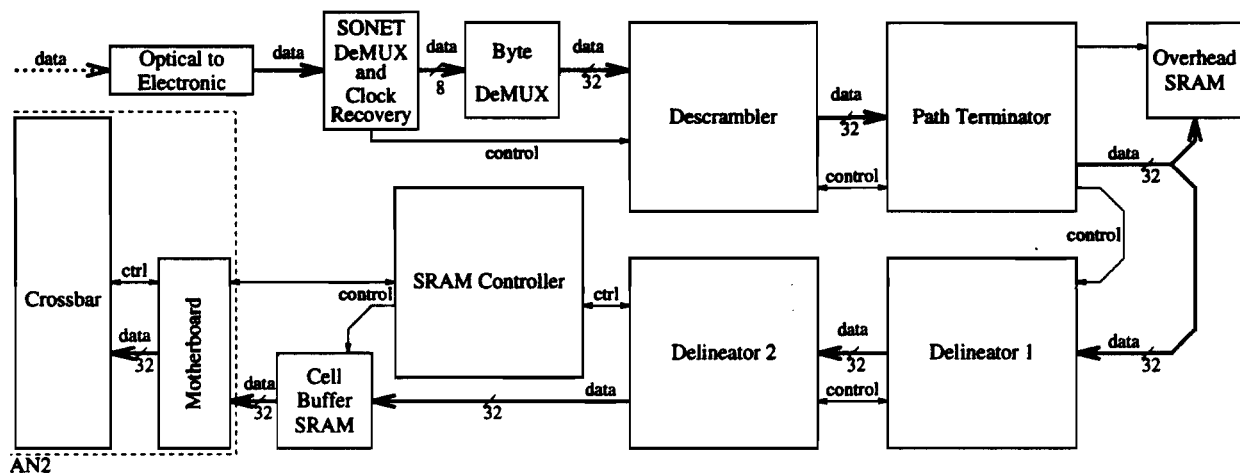


Figure 4.3 Block Diagram of Receive Section of Gateway

The second FPGA in the receive data path is the path terminator. This component determines the location of the start of the synchronous payload envelope (SPE) and the path overhead. It also provides the four least significant bits of address for the receive overhead SRAMs and the required control signals to store all of the overhead, including the path overhead.

The third and fourth FPGAs in the receive data path are the ATM delineation chips. The first component delineates the ATM cells out of the received stream. This is done by continuously checking for the HEC byte in the header and verifying whether it matches the CRC computed on the previous four bytes. The second chip performs ATM descrambling and acts as a partial cell buffer (required because complete cells cannot be buffered within the FPGA). Partial cells, in the form of 32-bit words, are buffered before being written to a buffer SRAM where cells are reconstructed.

The SRAM controller FPGA controls the cell reconstruction within the reconstruction SRAM. Partial cells from multiple streams can be multiplexed into the SRAM where complete cells are created. These cells are then forwarded to the AN2 for switching. This scheme in combination with link-by-link flow control guarantees the bandwidth available to each stream and prevents congestion and consequent cell dropping.

The one quality that is unique to the KU gateway is its richness of ATM support components. These include a traffic measurement device, a rate-based controller and associated measurement system, and a credit-based flow control implementation. A block diagram of the gateway support chips is shown in Figure 4.4.

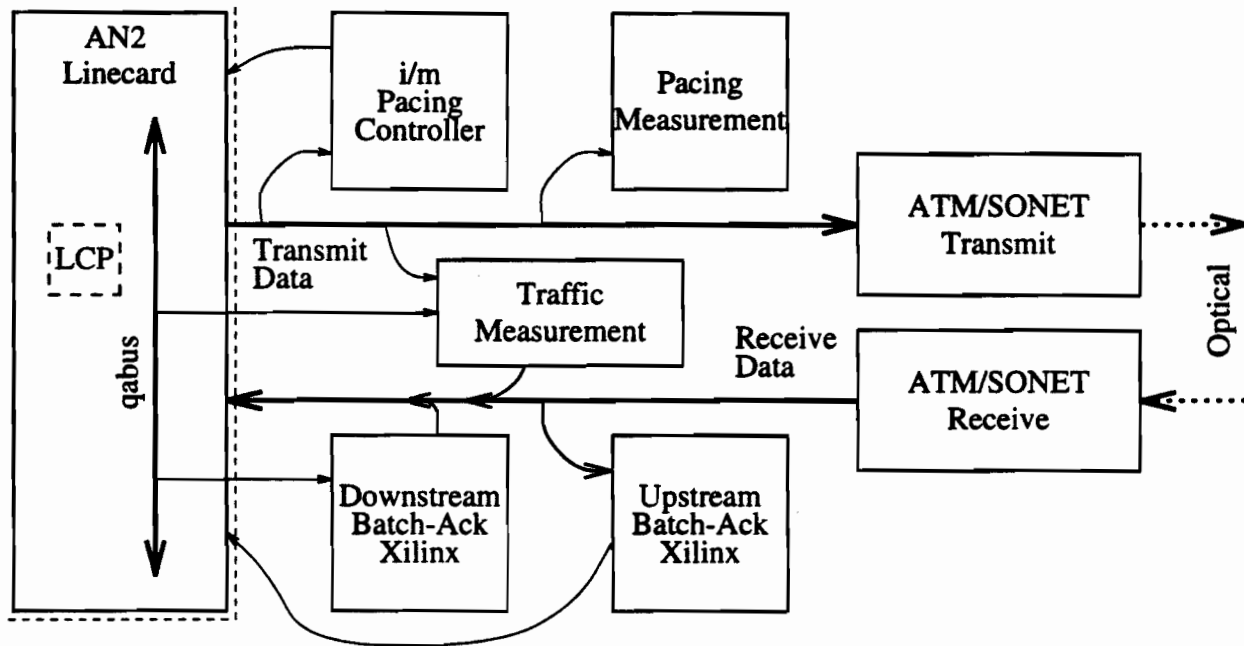


Figure 4.4 Block Diagram of ATM Support Chips in Gateway

The traffic measurement FPGA monitors the output and input port of the AN2 line card interface to the SONET gateway. The VCIs as well as the AAL5 payload type indicator (PTI) of the cells that are transmitted and received are sampled at every cell slot and packaged into ATM cells. Note the PTI can be used to reconstruct AAL5 traffic flows. Up to three streams that carry this information can be configured. The first stream carries the VCIs of the cells on the output in its payload. The second stream carries the VCIs of the cells going into the cell buffer on the linecard (common board). The third stream carries the VCIs of the cells emanating from the VRAM buffer on the linecard as they are being forwarded. These three streams are injected into the cell buffer SRAM in a manner similar to the ATM delineators. Currently, only two streams are implemented due to spatial restrictions in the FPGA.

The *i/m* controller FPGA implements rate-based flow control on the gateway. The algorithm implemented is a strict rate-based control (pacing) mechanism. It allows no more than *i* cells in any *m* continuous cell slots. Thus *i* is the maximum burst size and

the ratio i/m is the maximum long-term average rate that this particular VC can achieve. This is implemented on a per-VC basis, and any number of the 4000 VCs supported on the gateway can be simultaneously flow controlled independently of the other VCs. This has some implications, one being that the total bandwidth allotted to the VCs must not exceed the link capacity. Also, because the AN2 restricts any VC's maximum rate to 400 Mb/s (half of the capacity), the ratio i/m should not exceed 0.5 or $2/3$ of the SONET bandwidth. Tables of i and m are maintained on a per-VC basis. Within the hardware, the length of the m field is 10 bits while that of the i field is 9 bits. The values are set and modified by the LCP. One bit in the hardware is used to indicate whether that particular VC is to be paced or not. When a VC has temporarily exhausted its tokens (i value), it is stopped. That is, no more cells will be transmitted on this VC until its tokens are updated and the VC is started. Restoration of tokens is performed in hardware, m cell slots after that particular token was used. This pacing scheme is independent of the credit-based flow control on the gateway.

The pacing measurement FPGA is used to monitor the VCs that are paced. A maximum of 127 VCs can be monitored. Three fields are maintained on a per VC basis. A rate measurement provides the 24-bit running count of the number of cells transmitted on that VC. A single bit "stopped flag" indicates whether the VC is currently stopped. This bit remains set until a cell is transmitted on that VC. The third field is the burst count, which indicates the number of bursts in which the VC was active. A burst occurs when the line is idle, becomes active, and again goes idle. Bursts are defined by the line's being idle for two consecutive cell slots. These three fields can be read and cleared by the LCP and are used in conjunction with the i -out-of- m controller for dynamic bandwidth allocation.

The fourth and fifth support FPGAs implement the credit-based flow control scheme. This is an extension of the AN2 flow control mechanism. It is tailored for the WAN environment by placing credits in standard ATM cells. One chip implements the functionality for the upstream node, while the other chip implements the functionality for the downstream node.

The downstream chip monitors the buffer in the linecard. When cells are forwarded out of the cell buffer, thus freeing up space, credit entries are created and stored. Credit entries are needed because the ATM header cannot be used for cell credits (as is done in the AN2) in the WAN. These credit entries are then packaged into standard ATM cells and injected into the traffic going toward the upstream node. Each credit entry holds the VCI and a credit value. The WAN is also capable of remapping the VCI of the cells for purposes of switching. The credit value can credit from 1 to 16 buffer spaces. A maximum of 24 credit entries can be packaged into a single ATM cell. This method allows for "tunneling" through a network that does not implement credit-based flow control. Currently, the credit cell does not carry any error detection or error protection codes. Up to four VCs can be used to carry credits to the upstream nodes. These VCs may be switched in the network by non-flow controlled switches and routed to their destinations.

The upstream chip monitors the arriving cells as they are stored in the VRAM. When a VC is detected as one that carries the credits from any of the four downstream nodes, the payload of this cell is buffered. The credit entries are extracted from the cell and stored in a FIFO to prevent loss of credits if multiple cells arrive within a short period of time. The credit entry, one per cell cycle, is passed to the linecard by an interface that accepts the VCI and credit value. A summary of the features of the gateway is given in Table 4.1.

Modes	Flow Control	Measurements
4 x OC-3c	Static i-out-of- <i>m</i> rate control	VCI traces (VCI number vs. time slot)
OC-12c	Dynamic i-out-of- <i>m</i> rate control	AAL5 traces (AAL5 PDU start and stop times)
	WAN-based link-by-link credit flow control	

Table 4.1 Summary of the Features of the ATM/SONET Gateway

4.3 Gateway Results

Several experiments were conducted to verify the operation of the cell-level measurement capability and evaluate the throughput performance of the gateway. The traffic chip was used to gather data on an experiment between the Sprint Technology Integration and Operations Center (TIOC) and KU. Two separate experiments were performed. Both tests involved DEC Alpha stations with OC-3 OTTO cards at each site. The tool for measuring application layer throughput was *ttcp* (*ttcp* performs memory-to-memory copies using TCP over the network as fast as possible and measures the throughput). The first experiment was performed without the use of flow control (pacing). Depending on CPU load, between 170 and 1200 packets/sec were transmitted using a TCP window size of 64 Kbytes and a buffer size of 128 Kbytes. The traffic chip was used to observe the cell stream within the gateway. The mean cell level throughput was 142 Mb/s, near the theoretical maximum for this case and consistent with the observed application layer throughput. A histogram of cell interarrival times for this case is shown in Figure 4.5.

The second experiment was performed with flow control, i.e., with cell-level pacing. In this case the pacing flow control was set to allow 5 Mb/s or 67 packets/sec. Again, the chip was set up to sample the cell stream, and the observed mean cell level throughput was 5.3 Mb/s. Also there is some spread to the paced interarrival histogram (Figure 4.6) because the AN2 was not operating in a constant bit rate (CBR) mode. Data was gathered and averaged for about 18 ms in both cases. Figure 4.7 shows the effect of pacing on the cell arrival time series as observed by the measurement chip. These experiments confirmed that the measurement capability was operating properly.

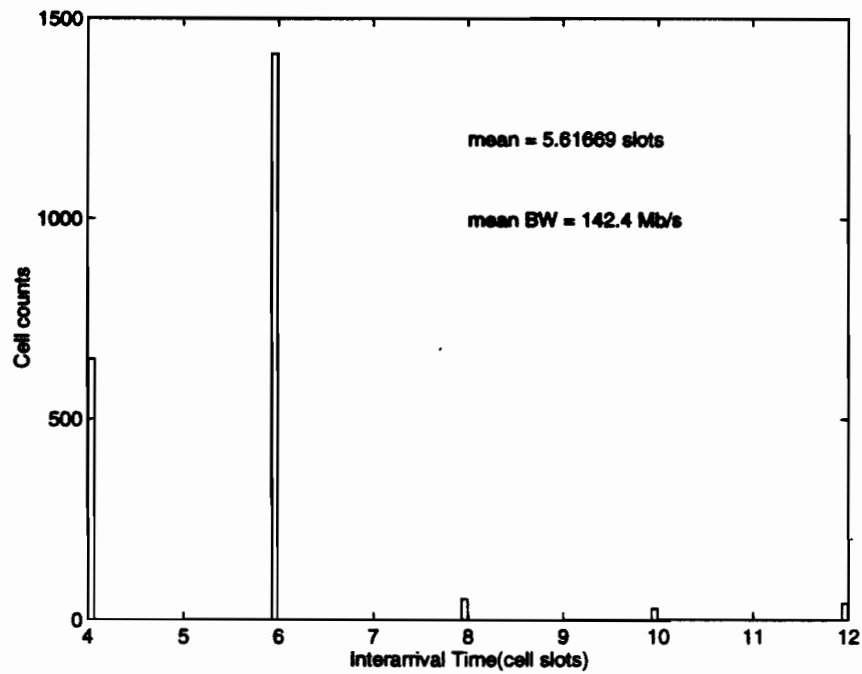


Figure 4.5 Histogram of Cell Interarrivals for Unpaced Stream

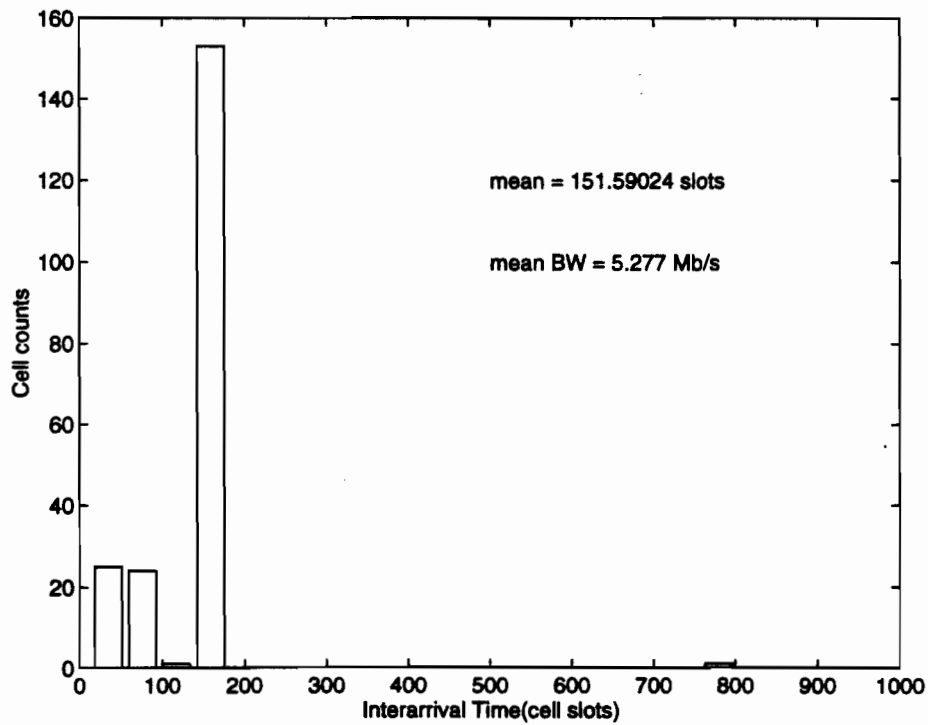


Figure 4.6 Histogram of Cell Interarrivals for Paced Stream

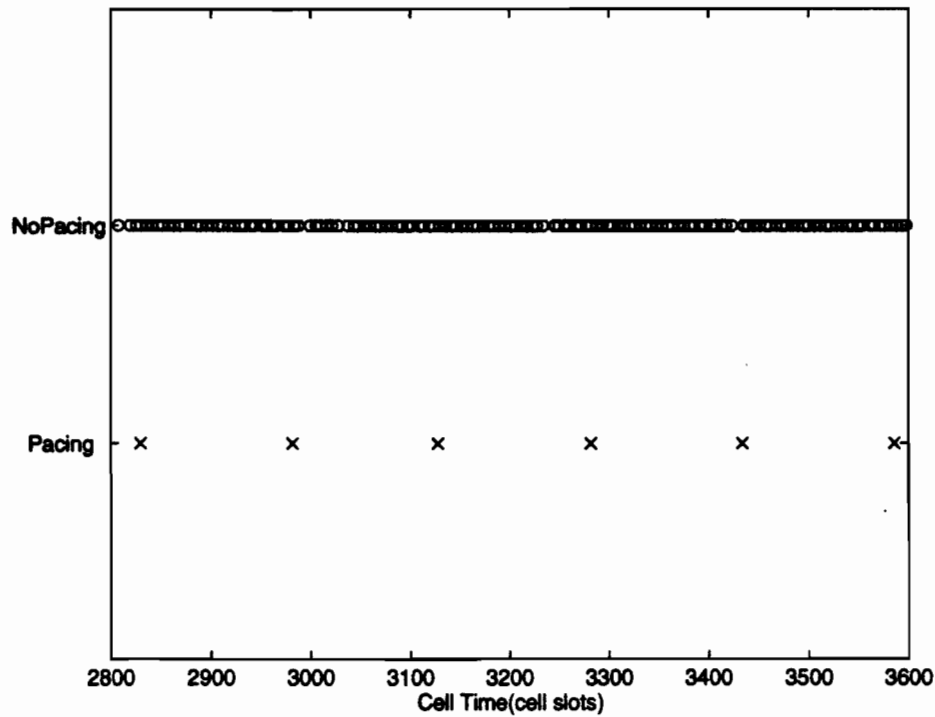


Figure 4.7 Time Series of Cell Arrivals for Paced and Unpaced Streams

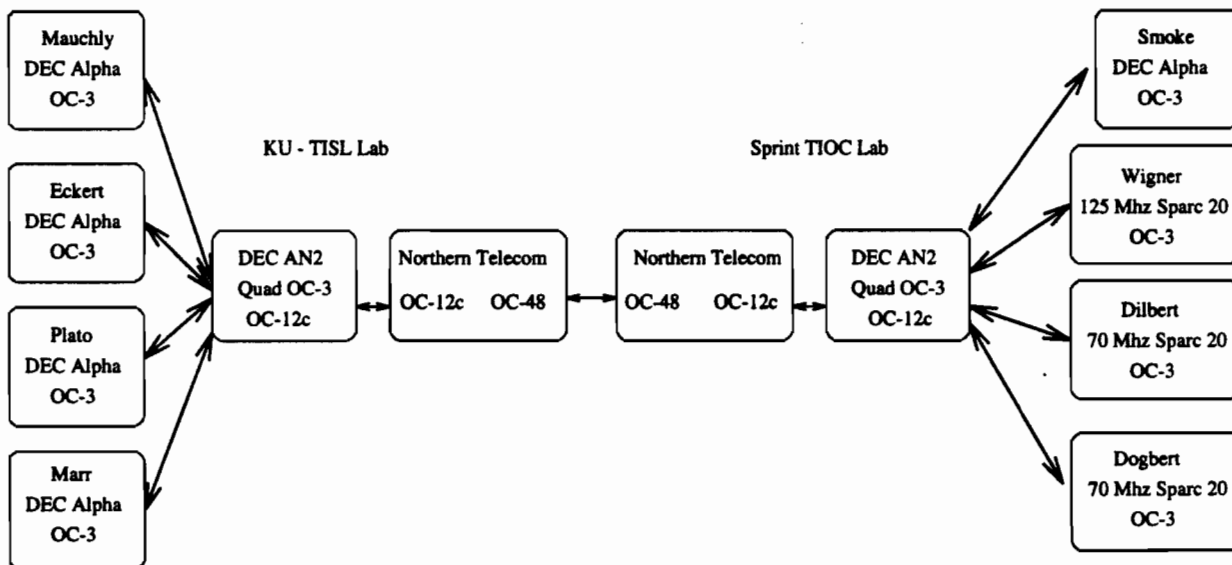


Figure 4.8 OC-12c WAN Experiment

An experiment was conducted between the Sprint TIOC and KU to evaluate the throughput performance of the gateway in OC-12c mode. This experiment used the configuration as shown in Figure 4.8. Four DEC Alpha workstations with OTTO OC-3

cards at KU sent traffic to a DEC Alpha, two 70Mhz SPARC 20s, and one 125 Mhz Dual SPARC 20, each with OC-3 cards. The AN2 at each site was connected via an OC-12c interface on the OC-48 fiber-optic terminal.

First an Alpha-to-Alpha stream and an Alpha-to-multiprocessor SPARC 20 stream were set up. The tcp tool was used with 64 Kbyte windows and 128 Kbyte buffers, and both streams were started at the same time. Observed throughput was about 110 Mb/s on the Alpha-to-Alpha stream and 95 Mb/s on the Alpha-to-Multi SPARC 20. Next the other two Alpha-to-70 Mhz SPARC 20 streams were set up. Table 4.2 contains the results when all four streams were sent at the same time.

Stream (Source - destination)	Throughput (Application Level)	Throughput (Physical Level)
Alpha - Alpha	92 Mb/s	124 Mb/s
Alpha - Multiprocessor SPARC 20	85 Mb/s	114 Mb/s
Alpha - SPARC 20	57 Mb/s	80 Mb/s
Alpha - SPARC 20	54 Mb/s	73 Mb/s
Total	288 Mb/s	391 Mb/s

Table 4.2 Gateway Throughput

A total of 288 Mb/s application-level throughput was obtained corresponding to a physical-level throughput of about 391 Mb/s. Packet errors were seen on all streams. One possible source of these errors was cell loss on non-flow-controlled streams. The observed throughput is consistent with buffer overflow problems on non-flow controlled streams. This problem is currently being investigated. It is expected that much better performance can be obtained when all streams are flow controlled. Currently, the hardware is not available to perform that experiment.

5.0 Real-Time Traffic Control in ATM WANs

Understanding the performance characteristics of ATM wide-area networks is of fundamental importance to the evolution of the national information infrastructure. The MAGIC network provided an early opportunity to gain insight into the characteristics of ATM WANs. In this section, the initial disappointing performance of the MAGIC network will be presented, followed by a discussion of a static traffic control algorithm that was employed to reach throughput performance close to the theoretical limits of the network. The design and real-time implementation of an adaptive traffic control algorithm for the AN2 gateway will also be described.

5.1 The Need for ATM-Layer Traffic Control in WANs

The initial measured performance of TCP/IP over the MAGIC wide-area SONET/ATM network was disappointing. An experiment was configured using a Digital Equipment Corporation DEC 3000 AXP (using an OC-3c interface) in Lawrence, Kansas, transmitting to a SPARCstation-10 (using a TAXI interface) at EDC in Sioux Falls, South Dakota (8.8 ms round-trip time). This configuration consists of a single host transmitting to another host with a 155 Mb/s (OC-3c) to 100 Mb/s (TAXI) bandwidth constriction in the path. The experimental conditions in this case included 128 kByte TCP windows and 64 kByte tcp write buffers. In this configuration, the throughput was only 870 kb/s, or less than 1% of the available capacity.

The performance with multiple traffic streams was also poor. In this case, two DEC 3000 AXPs (OC-3c) in Lawrence, Kansas, transmitted to a SPARCstation-10 (TAXI) in South Dakota (8.8 ms round-trip time) to evaluate the effect of two hosts causing contention on the switch port to a third host. The experimental conditions included 128 kB TCP windows, 64 kByte write buffers in tcp, and no ATM traffic control. The combined throughput was 1.6 Mb/s in this case.

It was clear from these measurement results that TCP rate control was not effective. Buffer overflows and subsequent packet retransmission resulted in poor performance. Therefore, some form of ATM-level traffic control was needed.

5.2 Static ATM-Layer Traffic Control for ATM WANs

Analysis of the results from these initial experiments indicated that rate mismatches, which will be common in ATM WANs, and switches with small buffers will result in poor throughput unless some form of ATM-layer traffic control is used. One form of such control, cell-level pacing, was developed here to significantly improve the performance of the MAGIC network.

Cell-level pacing is implemented on the DEC Turbochannel OC-3c (OTTO) interface. This pacing was originally intended for sending constant bit rate (CBR) traffic. Pacing

can be enabled on a per virtual circuit (VC) basis. It is implemented as a transmission schedule in which a time slot is allocated to a particular VC. If no traffic needs to be sent on that VC during that time slot, the slot is available for use by other VCs. For example, VC_i in Figure 5.1 could use up to half of the available bandwidth (its scheduled slot and up to one other i out of every four) if there is any traffic to send on that VC. The OTTO interface includes a hardware implementation of AAL5, which was used exclusively in these tests.

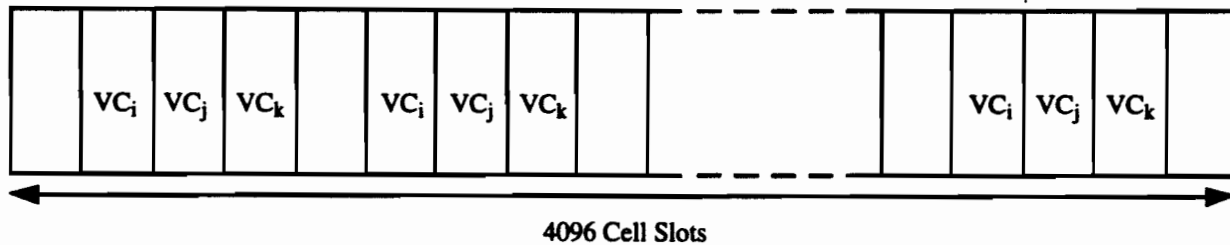


Figure 5.1 OTTO Pacing Scheme

With cell-level pacing (with a pacing rate of 70 Mb/s), the throughput went from 870 kb/s to 68.2 Mb/s for a single stream and from 1.6 Mb/s to 52.3 Mb/s for the multiple streams in the cases just described. (The reduced rate, from about 68 Mb/s with one stream to 52 Mb/s with two, is due to over head in processing two streams in the destination workstation.)

Clearly, ATM-level pacing significantly improved the performance of the MAGIC network. Further, these experiments increased the level of understanding of ATM WANs. See Appendix B for a complete description of the effects of traffic shaping on link utilization and congestion in ATM-based WANs.

5.3 Development, Design, and Implementation of Real-Time Traffic Control for ATM WANs

From our initial experience with the MAGIC network, it is clear that ATM-level traffic control is an important element in achieving high throughput in ATM WANs. In the above examples, the pacing rate was fixed by the user. A desirable property of LAN/WAN gateways would be to have this rate dynamically adjusted in response to traffic variations. It is also critical that this adaptive algorithm be implementable in real time. An algorithm with these characteristics has been developed for the AN2 LAN/WAN gateway.

The basis of the algorithm developed here is the i/m rate control discussed in Section 4.2 and reviewed briefly here. A user can transmit a maximum of i ATM cells in any m cell slot interval, yielding a maximum sustained throughput of i/m normalized to the line rate. The value of i also indicates the maximum number of consecutive cells that can be transmitted by that user. The real-time adaptive i -out-of- m shaper changes the i and m

values according to traffic measurements to match the rate (i/m) and burst (i) characteristics of the source.

There are many approaches to providing adaptive traffic control. An obvious approach to such control would be to monitor the output of the source and adapt to changes in the traffic submitted by the user to the network. However, it is not possible to directly monitor the source output from the AN2 ATM/SONET gateway. The only measurement that can be obtained is at the output of the round-robin queue on the gateway card, as seen in Figure 5.2. The traffic characteristics of the output of the round-robin queue differ from those of the source due to queueing effects and the contention arbitration performed by the DEC AN2. We found that the average rate and average burst dynamics on a per VC basis capture the actual source characteristics with sufficient fidelity to adapt the i -out-of- m shaper parameters. These values are provided by the pacing measurement FPGA discussed in Section 4.2.

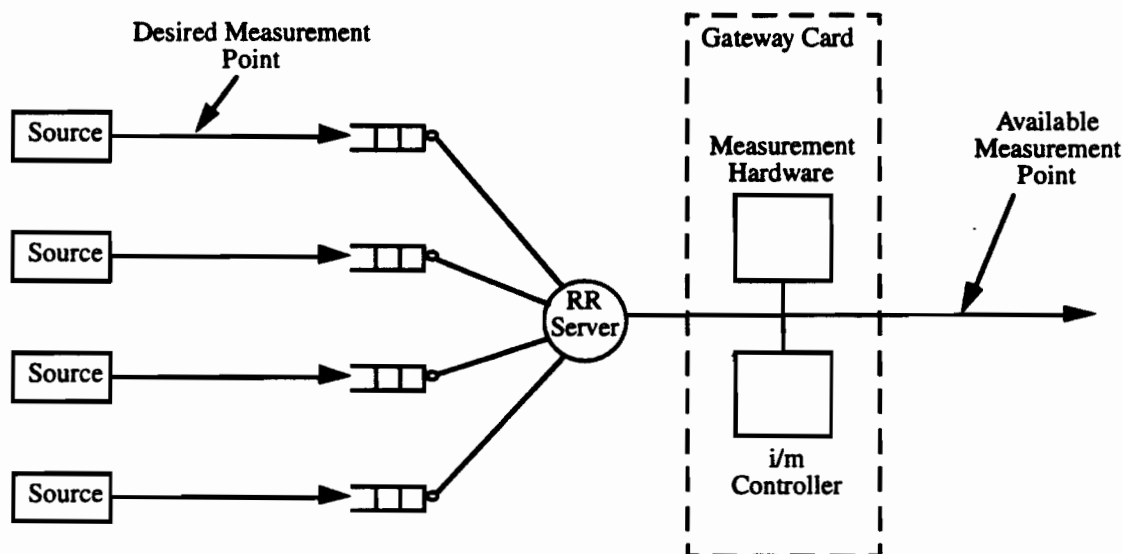


Figure 5.2 Measurement Points on the ATM/SONET Gateway Card

A running count of the number of cells transmitted on a particular VC in a known measurement interval was used to compute the average rate. The measurement hardware also provides the number of bursts that have occurred on a given VC in the measurement interval. This can be used in conjunction with the cell count to approximate the average number of cells in each burst (the average burst size):

$$\beta = \frac{\text{Total number of cells in a measurement period}}{\text{Number of bursts in a measure period}}$$

For a burst to occur, the line must initially be idle (no cells from any VC are being transmitted), then become active (at least one cell transmitted on any VC), and again go idle. If a VC transmits at least one cell during a burst, then the burst count for that VC is incremented. An exception to the burst parameter rule is the case of a single active

source. Due to a hardware restriction on the DEC AN2, a single source cannot transmit back-to-back cells. It can transmit, at maximum, during every other cell slot. According to the burst count mechanism described above, each single cell would be a burst. Clearly, this is an incorrect burst size if the source transmits more than one cell in a burst. To correct this problem, the VC number for the cell is compared to that of the last cell transmitted. If they are the same, the burst count for that VC is not incremented.

The adaptive mechanism considered here can be defined as a two-parameter control problem. The two parameters that need to be adapted are the average rate and the average burst size for a VC. In this algorithm the burst size adaptation is performed first and then the rate adaptation.

Figure 5.3 shows a flow chart of the algorithm. The values for i , m , and the measurement interval need to be initially set by the user. Raw measurements are obtained at the end of each measurement interval. From the measured rate (λ) and the present i and m values the rate utilization is calculated as

$$r = \frac{\lambda}{(i/m)}.$$

This parameter measures how closely the controller is estimating the actual rate of the source.

The algorithm begins adapting the burst parameter. This is done by comparing the value of i to the calculated average burst size. If i is less than $1.1*\beta$ then i should be increased since the number of tokens assigned to the VC is too small. Similarly, if i is greater than $1.3*\beta$ then i should be decreased since the number of tokens assigned to the VC is too large. If i is between 1.1 and 1.3 times the value of β , no adaptation is performed on i . Once the direction for the adaptation has been determined, the i adaptation is achieved using:

$$i = i + (i * \mu^+)$$

for an increase in the number of credits, or

$$i = i - (i * \mu^-)$$

for a decrease in the number of credits. The parameters μ^+ and μ^- are adaptation algorithm parameters.

When the burst adaptation has been completed, the rate adaptation is performed. Deciding the direction of the rate adaptation is simpler due to the lack of hysteresis in this adaptation factor. If the normalized rate utilization, r , is greater than a threshold (this is the target utilization), r_{targ} , the rate (i/m) is increased. However, if the rate is less than or equal to the target utilization, the rate is decreased. For validation of the algorithm, r_{targ} was set to 0.85.

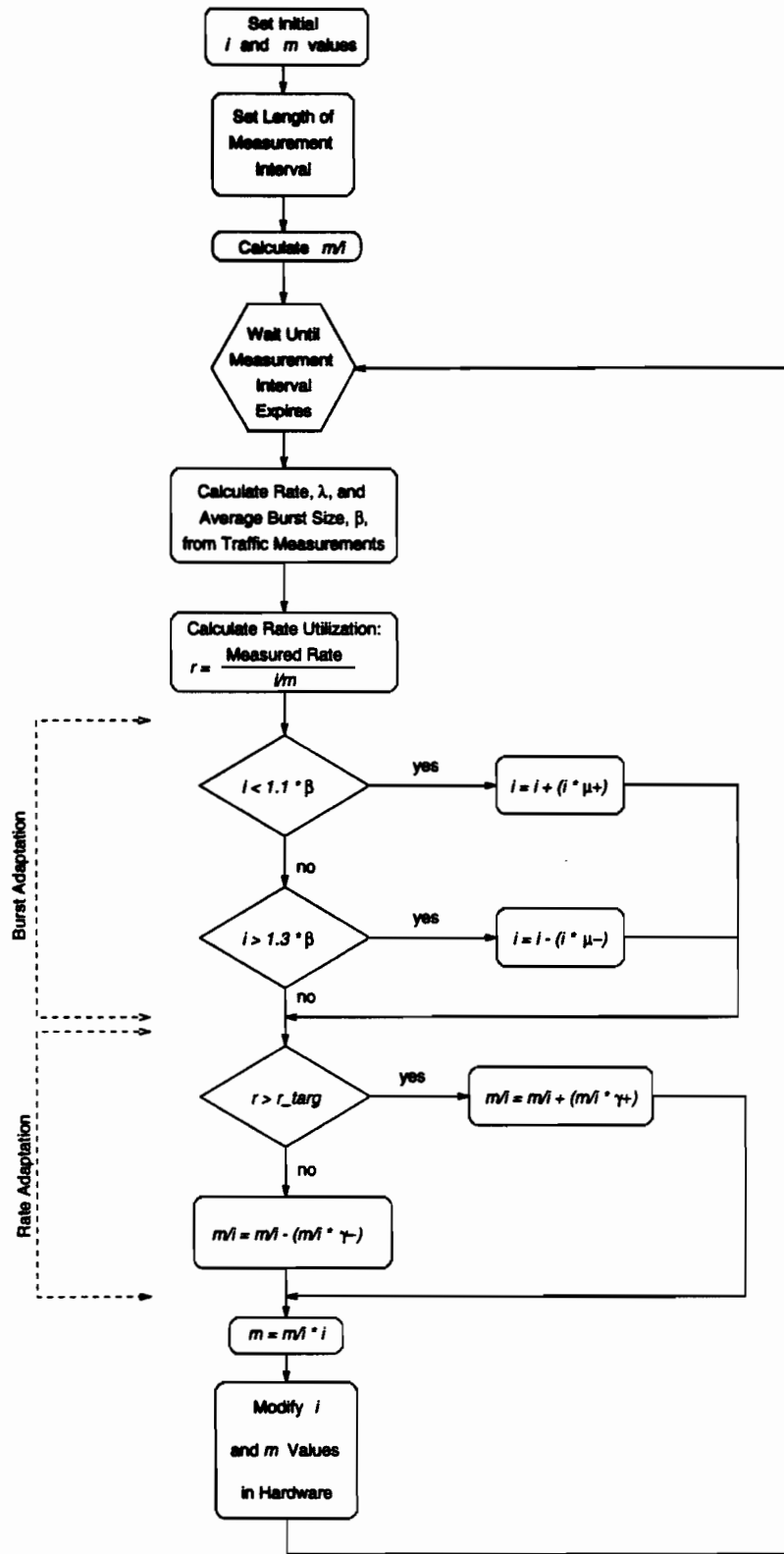


Figure 5.3 Adaptive i-out-of-m Shaping Algorithm

The algorithm m/i actually adapts the inverse of the rate, so that the new value of m can be obtained via multiplication $\frac{m}{i} \cdot i$ rather than division $\frac{i}{i/m}$. Hence, the rate adaptation is performed according to the equations

$$m/i = m/i - (m/i * \gamma^-)$$

for an increase in the shaper rate, or

$$m/i = m/i + (m/i * \gamma^+)$$

for a decrease in the shaper rate. The parameters γ^+ and γ^- are the adaptive coefficients for rate.

Before a real-time implementation of this adaptation algorithm could be attempted, some benchmarking of the hardware supporting the algorithm was required to determine realistic measurement intervals. The analysis of the gateway line card processor (LCP) indicated that a measurement interval in the range of 10 ms would be possible.

A simulation of the algorithm was performed to determine its effectiveness in adapting to changes in measured inter-LAN traffic. The traffic trace used in this study is inter-LAN Ethernet traffic obtained from Bellcore and represents a reasonable source that might submit traffic to a WAN.

Figure 5.4 shows the results obtained from the simulations using the trace data. The data points on these figures are values averaged over a one-second interval. From Figure 5.4 it can be observed that the dynamic shaper adapts to the significant rate change. The average delay using the adaptive shaper with a measurement interval of 50,000 cell slots (26 ms) was 11 ms, which is much shorter than the 160 ms delay obtained from a conservative static assignment policy where i/m is fixed at 0.02. Additional performance results are contained in Appendix C along with details of the implementation of this algorithm in the AN2 LAN/WAN gateway.

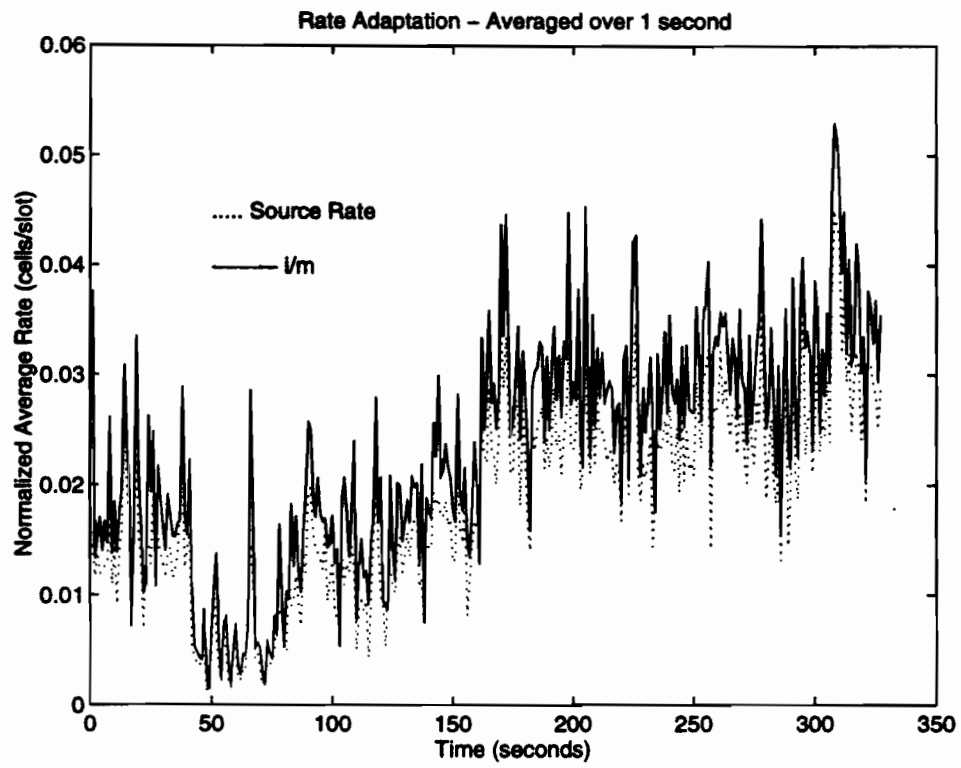


Figure 5.4 Rate Adaptation of Shaper to Ethernet Trace Data

6.0 The Over-The-Shoulder Application

Over The Shoulder (OTS) is a software package that is designed to allow for the remote viewing and control of XWindows-based workstations. It is a client/server system that consists of two clients and two servers. Through the use of these programs, the user can view low-resolution images (miniatures) of up to 12 remote workstations simultaneously, view and control one remote workstation at its actual (high) resolution, and distribute these images for viewing by others across a multicast connection in real time. OTS was created to give the user several specific capabilities not already available in other software packages as well as to serve as a test application for high bandwidth networks.

The first feature of OTS allows a user on workstation A to view in real time exactly what the user on workstation B is viewing. This feature was originally intended for use with the TerraVision application. It would allow a supervisor at a remote site to see what the person controlling TerraVision was doing without actually being in the same room or even the same state. This feature would also be of use to a member of a technical support staff. The technical staff member could use OTS to help a user correct problems without having to ask many lengthy and possibly confusing questions.

The ability to control the remote workstation (without the need of an account) would also be of use to technical support staff. The support person could show a user how to solve a problem without either person ever leaving their offices. This feature can also be used to log into the remote machine (assuming the user has an account there) and set up a demonstration without having to be there physically. The advantage of this method over simply telnetting into the remote machine is that not only can the user "see" that things are working as they should, but many programs can only be controlled through an XWindows interface and thus not set up appropriately without the user sitting at the machine. This particular situation is very common. There are security features built into OTS that allow the owner of the machine being viewed to prevent unwanted outside sources from taking over the local machine.

OTS also allows laboratory monitors, system administrators, and others the ability to view miniatures of multiple screens simultaneously. This feature permits the monitoring of entire laboratories even if these laboratories are spread over a variety of rooms and buildings. This feature is not very demanding of either the CPU or computer network and thus can be used effectively over connections as slow as Ethernet (OTS is designed to work best over high bandwidth networks such as those available with ATM).

The capability to record and log the network usage of OTS is the feature most useful to network researchers. Data can be taken at the read/write system call level and later transformed into throughput and interarrival characteristics. Thus, OTS is an ideal tool for determining the ability of network configurations and protocols to operate under real world conditions. In addition to all of the above, OTS is capable of transmitting the image data that it acquires via multicast. This would be useful when teaching a

computers class and allowing all of the students to see what the teacher is doing on his or her own workstation. It is also convenient to use OTS in conjunction with other multicast services when doing remote conferencing, etc. The advantage of OTS over current products that perform similar functions is that OTS not only gives the user a miniature of the entire remote screen, but is capable of simultaneously displaying variable sized portions of the screen. Current products only give the user a fixed size view of part of the remote screen. Depending upon the user's needs, the above functions can be used individually or simultaneously.

6.1 Architecture of the OTS Application

OTS is designed as a client/server system. It consists of a Master Client, Slave Client, HighRes Server, and LowRes Server, as shown in Figure 6.1. Both servers are designed to allow simultaneous connections from multiple remote clients. The code is constructed so that it will run on any platform that supports XWindows and UNIX sockets.

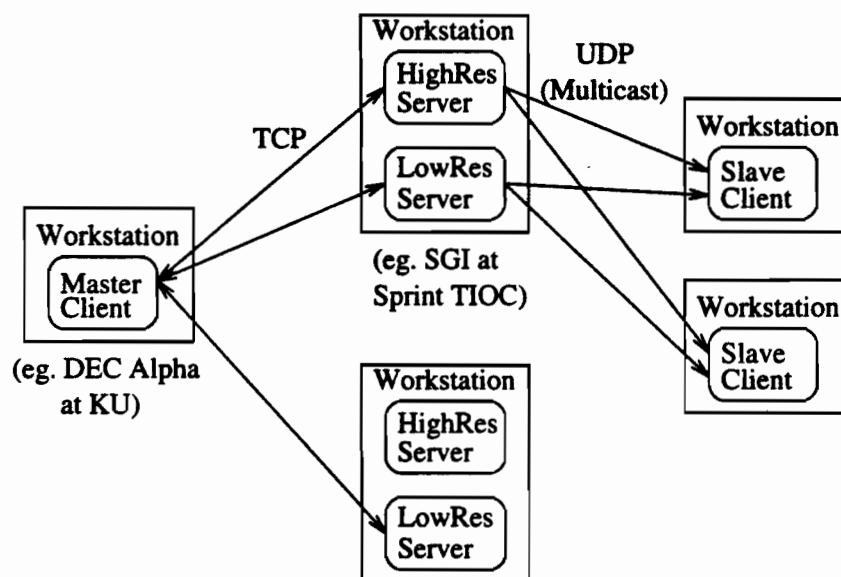


Figure 6.1 Client/Server Architecture for the OTS Application

The HighRes server has two main functions. First, it sends high-resolution images to the master client (and to the slave client if multicast is enabled). Second, it translates and executes any remote button presses or key presses sent by the master client. It also gathers performance information, which it sends back to the master client for processing in real time.

The LowRes server only has one purpose, to send miniatures of the entire screen to the

requesting clients. These images can range in size from 1/8 to 1/2 of the original screen size. Both the LowRes server and the HighRes server can be run manually or installed so that they are running whenever the workstation is booted.

The master client is where the controlling user sits. From here the user can select which remote workstations to view, what size the remote images should be, and at what rate the images should be transmitted. This program can observe up to 12 remote screens at one time and can take remote control of one of them. It is from here that all operational decisions are made. These include whether or not to transmit the images via multicast, which 24-to-8-bit conversion process should be used, and what mode the remote servers run in (no remote control, remote control only, or dual control by both local and remote user).

The slave client is simply a multicast receiver that knows the OTS communication protocol. It will watch a user-selected multicast address and grab and display any images that are sent by either of the server types. The Slave Client never 'connects' to any other program and thus has no control capabilities.

6.2 Performance of the OTS Application

After OTS was completed, several tests were conducted to determine its performance over an ATM/SONET network. In its various modes of operation under several different computer architectures, OTS obtained a maximum network bandwidth of around 55 Mbit/s. In normal use OTS required about 12 Mbit/s. OTS was tested over the MAGIC network between KU and Sprint in order to demonstrate the effects of congestion bottlenecks in an ATM WAN and to get some initial traffic characterizations. A total of ten experiments were run using the following network elements (see Figure 6.2 for the network configuration and Figure 6.3 for the OTS protocol architecture):

- OTS server running on an SGI Onyx at the Sprint TIOC in Overland Park, Kansas;
- OTS client running on DEC Alpha 3000/400 workstation at TISL in Lawrence, Kansas;
- cross traffic source was from a Sparc 20 workstation at the TIOC;
- cross traffic receiver was a DEC Alpha 3000/400 at TISL;
- a FORE switch at the TIOC connected to a DEC AN2 switch at TISL was used to complete the connection.

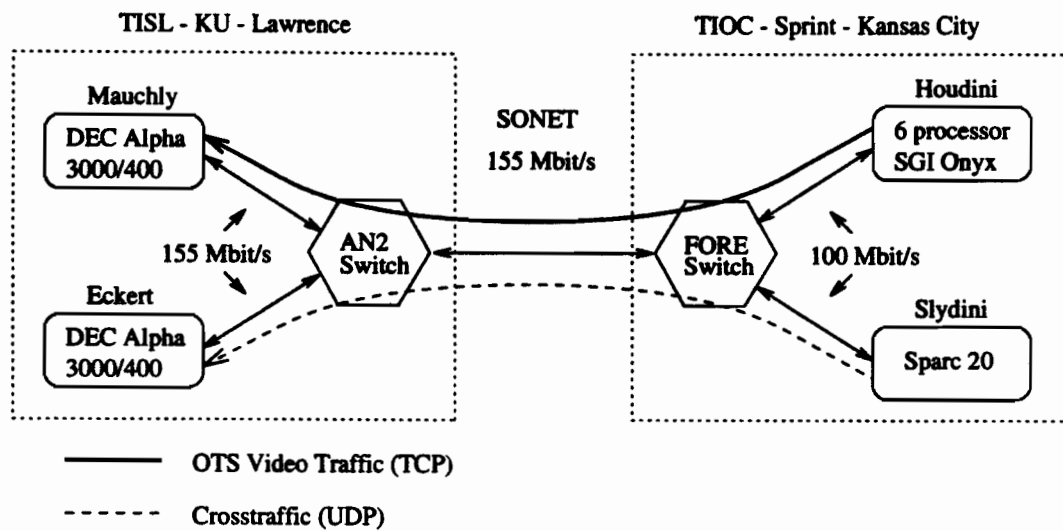


Figure 6.2 Configuration of OTS Experiments

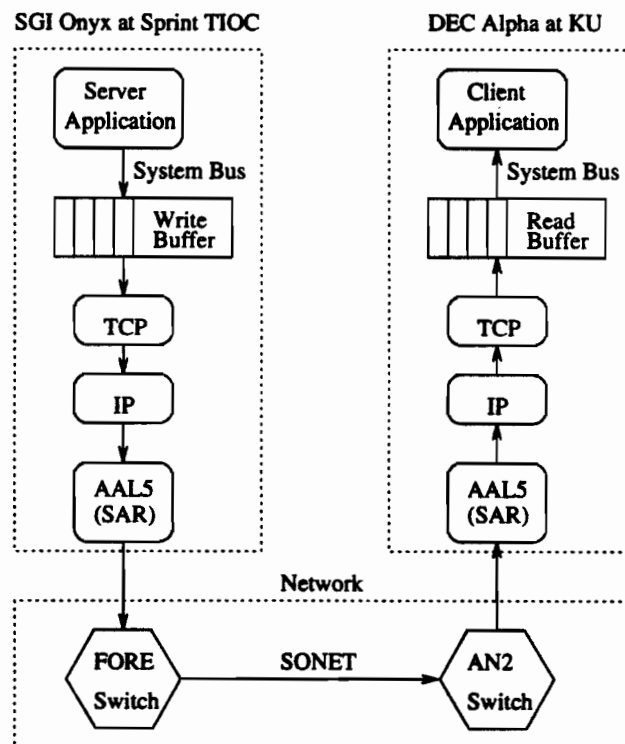


Figure 6.3 Protocol Architecture

The measurements were taken at the application level on both the receive and transmit sides, while measurements at the AAL5 level using ATMTIMES (see Section 8 for a discussion of ATMTIMES) were taken on the receive side only (tools to measure AAL5 flows were not available for the SGI Onyx). All cross traffic was generated using the TTPC program with UDP protocols. Below is a description of each of the ten experiments run using the above configuration.

Case 1: The HighRes frame rate was set to maximum (as fast as the workstation would allow) and no LowRes frames were sent.

Case 2: The HighRes rate was set to maximum with the LowRes frames being sent at a period of one frame every two seconds.

Case 3: Same as Case 2, but with user interaction (a user was activating remote XWindows events such as mouse movements and button pushes through the OTS remote control feature throughout the test).

Case 4: HighRes was sent at 10 frames/s, LowRes at 2 frames/s, with user interaction (these are the default OTS rates that most users would use during actual OTS operation).

Case 5: Same as Case 4, but with a remote video application running (an MPEG player in this case). This test was performed to see if any of the traffic characteristics were altered by extra hits to the remote frame buffer.

Cases 6-10: Identical to Cases 1-5, but cross traffic was added.

Table 6.1 shows the throughput results of these tests.

Without Cross Traffic (Mb/s)	With Cross Traffic (Mb/s)
Case 1: 34	Case 6: 1.2
Case 2: 32	Case 7: 1.4
Case 3: 24	Case 8: 1.4
Case 4: 12	Case 9: 1.4
Case 5: 12	Case10: 1.2

Table 6.1 Effects of Cross Traffic on the OTS Application

Note that in Cases 6-10 the UDP cross traffic received an average throughput of 82 Mb/s. The dramatic decrease in throughput is due to small switch buffers combined with no cell-level flow control. Typical traffic patterns with and without UDP cross traffic are shown in Figures 6.4-6.9. Figures 6.4 and 6.5 illustrate the application level (frame) traffic arrival rate characteristics at the client as a function of time. Note the dramatic effect of cross traffic on the application. Histograms of the frame interarrival

times at the client are shown in Figures 6.6 and 6.7. Traffic was also observed at the AAL5 level. Figures 6.8 and 6.9 show the histograms for the AAL5 protocol data units with and without cross traffic, respectively. The cross traffic increases to variance of the AAL5 interarrival times, but the effect of congestion is not as dramatic in this case.

6.3 OTS: Lessons Learned

After designing and experimenting with OTS the following conclusions can be drawn. See [8] for a detailed discussion of OTS as well as additional background for these conclusions.

- The traffic generated by OTS has similar characteristics on both the transmit and receive sides even across a WAN.
- ATM switches should be designed with large buffers to prevent data loss. Switch performance rapidly became unacceptable with only a small amount of cross traffic (even with flow control activated).
- When executing high-volume, high-speed communications over a WAN, system performance is directly proportional to the size of the I/O buffers on both the transmitting and receiving machines.
- Flow control is a necessity when using connection oriented protocols, such as TCP, over a loaded ATM network.
- On the operating system side, efficiency could be drastically increased if the maximum TCP packet size were increased to the maximum AAL5 packet size. Also, the current implementation of the read system call is very inefficient for large data transfers.
- When multiple images are to be displayed on a screen under XWindows, the use of shared memory is a necessity (it more than doubles throughput).
- OTS application traffic is very bursty.
- OTS application traffic does not have exponentially distributed message or interarrival times.

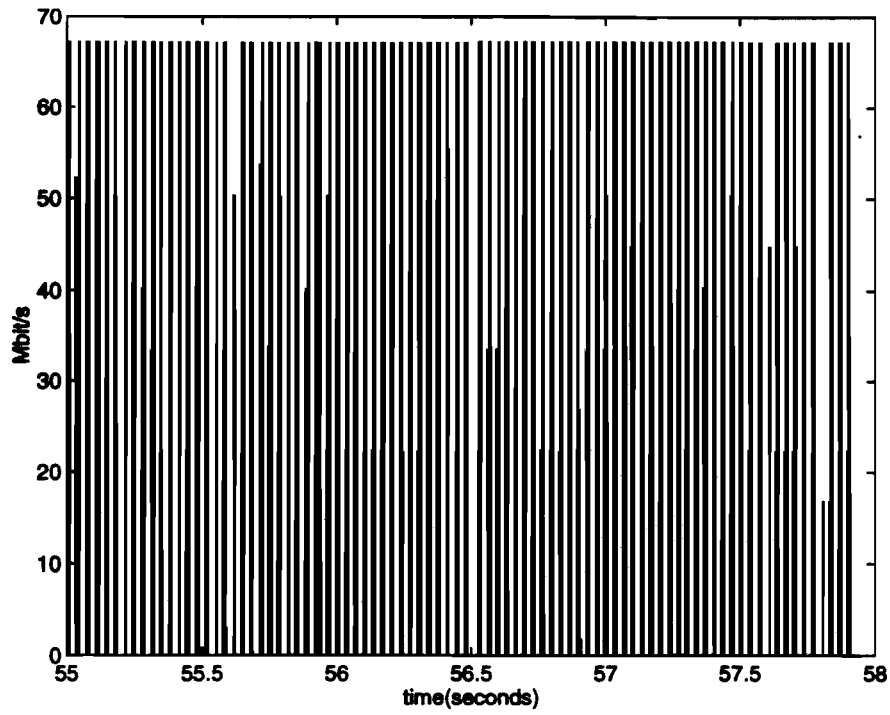


Figure 6.4 Frame Traffic Without Congestion

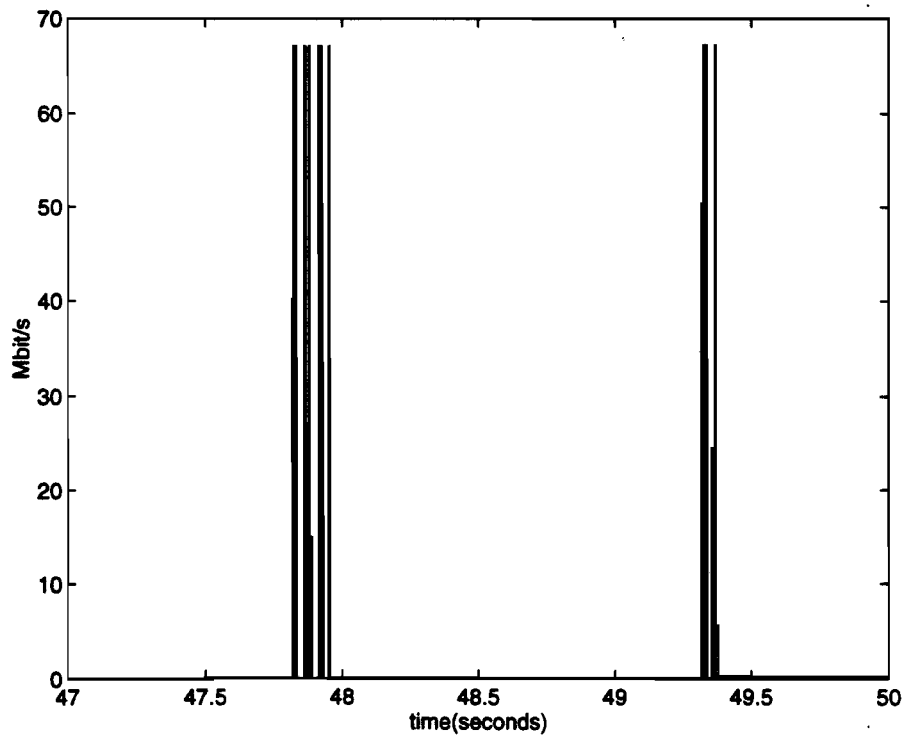


Figure 6.5 Frame Traffic With Congestion

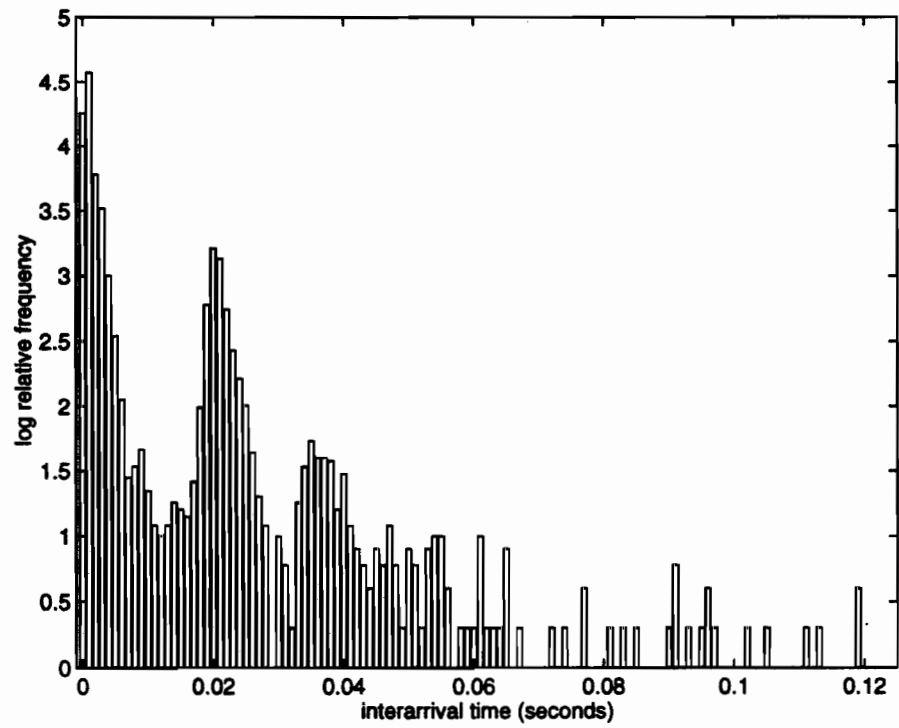


Figure 6.6 Histogram of Frame Interarrival Times Without Congestion

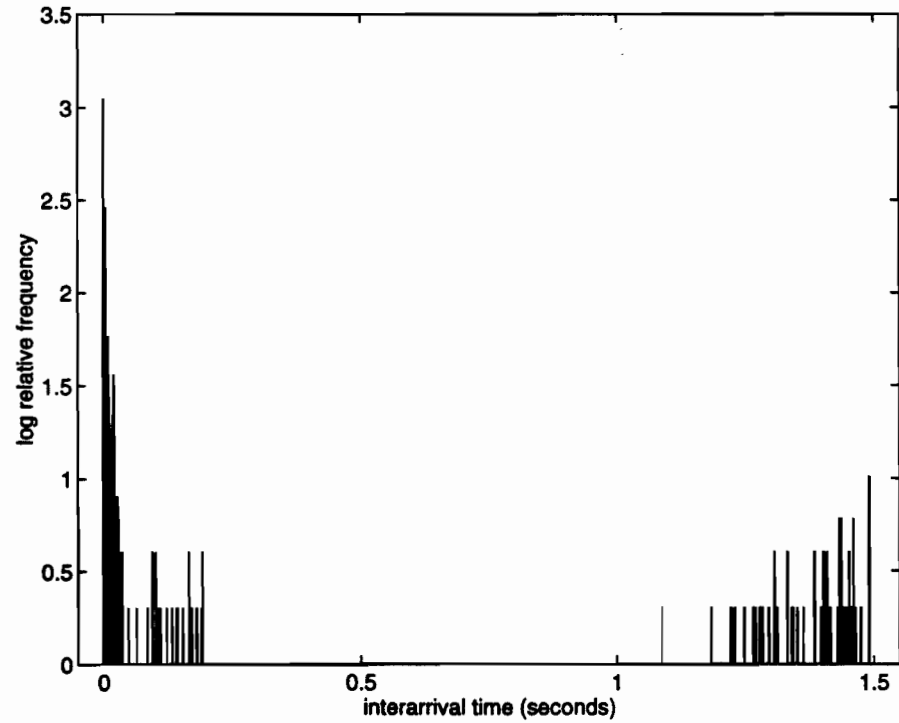


Figure 6.7 OTS: Histogram of Frame Interarrival Times With Congestion

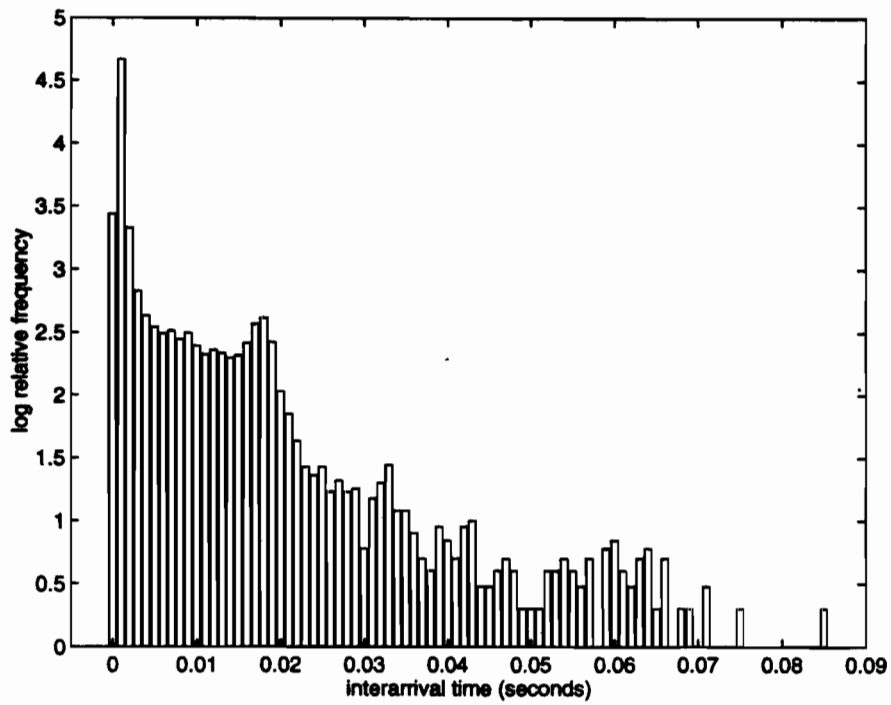


Figure 6.8 OTS: Histogram of AAL5 PDU Interarrival Times Without Congestion

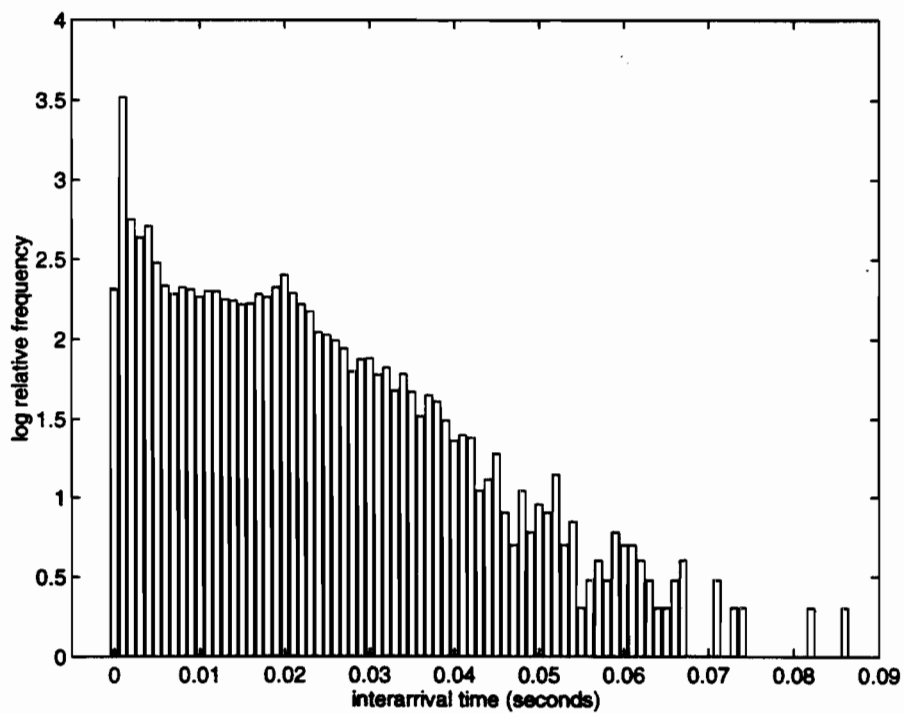


Figure 6.9 OTS: Histogram of AAL5 PDU Interarrival Times With Congestion

7.0 A Distributed Network Emulation System

Emulation of communications networks is an indispensable tool for the research and development of system hardware and software. Designers of distributed networked systems require the ability to study the system operation under a variety of scenarios. For instance, designers of a routing protocol need to verify the functional features of the protocol as well as to analyze the performance features of the mechanism during normal operation and critical operation (e.g. link or node failures). Similarly, hardware designers of new network components need to study the functionality and performance of the equipment while the hardware is still under development. Gigabit networks like MAGIC will have a revolutionary impact on an important number of applications (e.g. imaging and distributed computing) because the network bottleneck will be eliminated. In parallel to high-speed technology, research into new applications can be developed and tested in a simulated high-speed network environment. A novel simulation package has been developed that provides for the development and testing of network applications and protocols prior to the implementation of the actual network. A network management framework is integrated into this package.

Traditional strategies to simulation typically involve a language-based approach (e.g. Simula or Simgen) or a model-based approach (e.g. BONeS). Both approaches, however, suffer from having simulation and testing separated from design and implementation issues. In this sense, an environment-based approach presents a more general solution to the problem of network simulation. A network emulation using object-oriented techniques for modeling of network objects in an environment-based approach provides a better model of the system under study. By emulating a network, real data is sent and received across the simulated network. In this way, rather than testing a system with data characterized by a designated probability distribution (e.g. Poisson), a system can be analyzed with data that is actually generated by the real applications to be used in the real network. This approach presents two advantages. First, the functionality of network components (e.g. host adapters and switches) and protocols can be tested and evaluated. Second, better models of the target network can be obtained since real data is injected into network scenarios.

The DIStributed Network Emulation System (DISNES) is a software package that allows the emulation of arbitrary network topologies using a discrete-event scheduling mechanism. Distributed emulation of a system involves a distributed system of processes that communicate with each other solely via messages. In our case, even though the core of the emulation runs in a single process, external distributed processes can be linked to DISNES to allow the study of distributed network protocols and applications.

Written in Modula-3, the DISNES software is embedded within a standard UNIX environment. By implementing an object-oriented programming (OOP) approach, DISNES can easily model the functionality of different network components or objects such as host adapters and switches. Relevant statistics associated with these network objects can be collected during the emulation period, thus allowing performance

evaluation of network components with external real data. Additionally, DISNES features a model for handling network management information through standard network management operations. In particular, DISNES includes a Management Information Base (MIB) that contains information associated with network components that can be handled through standard network management operations based on the Simple Network Management Protocol (SNMP). Finally, DISNES can perform dynamic network configuration of logical connections in the target network through standard SNMP operations.

DISNES is based on the AN2 simulator software originally developed by Digital Equipment Corporation's Systems Research Center, and it was modified and enhanced to display more powerful and general capabilities. Details about DISNES can be found in Appendix D.

8.0 An Overview of Tools to Support Gigabit Network Performance Evaluation

To evaluate the effectiveness of traffic control algorithms, as well as ATM WANs in general, new measurement tools were developed. These tools were designed to provide a correlated view of the characteristics of the network from the application through the AAL levels. One set of tools was developed to collect the data and another set to analyze it. Cell-level measurement capabilities were described in Section 4.

Measurements at the application layer focused on the nature of requests for network service and the times for those requests to be satisfied. Two applications, TTCP and the over-the-shoulder (OTS) remote screen viewing application developed by KU for the MAGIC project, have been modified to collect measurements. For the OTS application, frame delay, frame size, and frame interarrival times have been measured at the application layer. (See Section 6 for examples) The techniques used to modify the OTS and TTCP applications can be extended to other applications. These modifications permit the measurement of interarrival time (time between reads/writes to the network), frame sizes (in bytes), and frame delays (the delay of application layer between two measurement points) for the OTS application.

TTCP (created at the US Army Ballistics Lab), a benchmarking tool for determining TCP and UDP throughput performance, was modified to create a new program called `ttcptimes`. The `ttcptimes` utility added a timestamping feature in the application layer. Since the TTCP program tries to send traffic as fast as it can, this timestamp feature in both transmitter and receiver sides is used to evaluate the delay jitter performance of the transmitting host and the network. A tool to measure AAL5 traffic at the host has been developed. This tool is called ATMTIMES. Both AAL5 protocol data unit (PDU) size and interarrival times can be measured with ATMTIMES.

Another tool (Netalyse) was developed to store the data collected from the network, analyze the data, and present the results graphically. This tool allows for the convenient analysis of network measurements. A detailed description of this tool is contained in [7]. Examples of the use of these tools are found in Appendix B and [8].

9.0 Lessons Learned

Much was learned about the properties, performance, and limitations of a diverse internetwork containing gigabit ATM LAN and WAN systems. Specifically:

- High throughput is possible over wide-area TCP/IP-ATM networks, e.g. about 120 Mb/s to TCP using OC-3c links.
- To achieve high throughputs over wide-area TCP/IP-ATM networks, appropriate flow control and proper end systems configuration are essential.
- This work offered the first opportunity to experimentally verify that TCP traffic control will not cope with cell-level congestion for high-speed wide-area networks.
- Cell-level pacing was experimentally shown to be an effective scheme for coping with cell-level congestion in high-speed wide-area networks. Cell-level pacing is an open loop mechanism.
- Closed loop control, in this case hop-by-hop credit-based flow control, was experimentally shown to be effective for cell-level congestion management in the MAGIC network. See [9] for details of the performance of closed loop control over MAGIC.
- In uncontrolled LAN/WAN internetworks, i.e., networks without cell level control, it was demonstrated that non-courteous flows, e.g., generated by UDP, can negatively impact the performance of other users.
- The feasibility of implementing OC-12 switch interfaces with advanced traffic control and measurement functions was demonstrated.
- The feasibility of implementing adaptive cell-level traffic shaping at high rates, in this case 622 Mb/s, was demonstrated.
- Sophisticated dynamic bandwidth control mechanisms are promising but require careful implementations that take real-time programming issues into account.
- Interactive visual applications, like the OTS, generate very bursty traffic at both the application and AAL5 levels. This traffic is non-exponential.
- Small prototypes of critical hardware components, in this case the 622 Mb/s section of the gateways card, should be constructed and tested early in the development program.
- Hardware development with FPGAs should be treated as software projects.
- Network configuration using permanent virtual circuits is difficult and time

consuming. The success of ATM depends upon the introduction of switched virtual circuit capabilities.

10.0 References

- [1] Richer, I., B. Fuller, "An Overview of the MAGIC Project," M93B0000173, The MITRE Corp., Bedford, MA, Dec. 1, 1993.
- [2] 1994 MAGIC Technical Symposium, August 17-18, Telecommunications and Information Sciences Laboratory, University of Kansas, Lawrence, KS.
- [3] 1995 MAGIC Technical Symposium, August 1-2, Minneapolis, MN.
- [4] Leclerc, Y.G. and S.Q. Lau, Jr., "TerraVision: A Terrain Visualization System," SRI International, Technical Note #540, Menlo Park, CA, 1994.
- [5] Tierney, B., and W.E. Johnston, "Distributed Parallel Data Storage Systems: A Scalable Approach to High Speed Image Servers," ACM Multimedia '94, October 1994.
- [6] Goguen, M., "AN2: A Self-configuring Local ATM Network," in Proc. Nat. Commun. Forum (NCF), 1992.
- [7] Laan, R.J.W, et. al., "Netalyse: A Network Analysis Tool," University of Kansas Telecommunications and Information Sciences Laboratory Technical Report #TISL-1660-11, November 1994.
- [8] Swink, M.T., "OTS: An Enabler for Real-Time Collaboration over High-Speed Networks," August 1995, TISL Technical Report 9770-36.
- [9] Zhu, H., L.A. DaSilva, J.B. Evans, F.S. Frost, "Performance Evaluation of Congestion Control Mechanisms in ATM Networks," 20th International Conference of the Computer Measurement Group (CMG '95), December 3-8, 1995, Nashville, Tennessee.

Appendix A

Minden, G.J., J.B. Evans, D.W. Petr, V.S. Frost, "An ATM WAN/LAN Gateway Architecture," *Second IEEE Symp. High Perf. Dist. Comp.*, July 1993.

