

SAND--97-856/C

An Experimental Study of VBR Video over Various ATM
Switch Architectures¹

CONF-9704202--

Rose P. Tsang

corresponding author -
email: rtsang@ca.sandia.gov
phone: 510-294-3200

RECEIVED

APR 30 1997

OSTI

Infrastructure and Networking Research
Distributed Information Technologies Center
MailStop 9011
Sandia National Laboratories
Livermore, CA 94551

Jenwei Hsieh and David H.C. Du

Distributed Multimedia Center
Computer Science Department
University of Minnesota
Minneapolis, MN 55455

Abstract

One of the most important components of an Asynchronous Transfer Mode (ATM) network is the switch. Switch design is not a part of the ATM standards so vendors use a wide variety of techniques to build their switches. In this paper, we present experimental results of switching and multiplexing real-time Variable Bit Rate (VBR) video traffic (JPEG, MPEG-1, and MPEG-2) through two different ATM switch architectures. Real-time VBR traffic, such as digital video, is particularly interesting due to its high demands in terms of bandwidth, real-time delivery and processing requirements. Our experiments show that the 'fastest' switches, i.e., lowest latencies, do not necessarily perform better when transmitting VBR video. The impact of the high speed network components' characteristics, such as switch fabric architecture, buffering strategies, and higher layer transport protocols (i.e., UDP, TCP/IP), are illustrated through the experimental results.

Keywords: Asynchronous Transfer Mode (ATM), digital coded video, JPEG, MPEG, experiments, switch architectures, buffering strategies.

¹This work is supported in part by the (i) US Department of Energy under contract DE-AC04-94AL8500, (ii) the Advanced Research Projects Agency (ARPA) through AFB, contract number F19628-94-C-0044, U S WEST Communications, and the (iii) Distributed Multimedia Center (DMC), sponsored by US WEST, Honeywell, IVI Publishing, Computing Devices International and Network Systems Corporation.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

1 Introduction

One of the most important components of an Asynchronous Transfer Mode (ATM) network is the switch. An ATM switch must accept asynchronous and synchronous traffic, as well as connection-oriented and connection-less traffic. The speed at which a switch relays cells within a pre-specified cell loss probability bound is considered crucial; queueing and switching delay must be minimized. The switch processor must also be capable of handling high speed ports and aggregate throughputs of at least a couple gigabits per second.

Do high switch throughputs and low switch latencies translate into superior performance for all traffic types ? The ATM standard is expected to serve as the transport mode for a wide spectrum of traffic types with varying performance requirements. In this study, we are interested in how well an ATM switch can support can support real-time Variable Bit Rate (VBR) traffic. This is particularly interesting because a significant motivating factor in the underlying philosophy of ATM (e.g., fast packet switching, efficient usage of resources via statistical multiplexing) was the intention to efficiently support difficult traffic types such as VBR real-time and non-real-time traffic.

We present experimental performance results of real-time VBR traffic, namely digital compressed video, over two different ATM switch architectures. The experimental results are followed by a discussion of the results in relation to the capabilities and features of the switch architectures and network environment. We are interested in digital video because it is expected to be the predominant type of multimedia traffic transmitted on future high speed networks [3, 6, 19]. Video requires the continuous periodic delivery of data in order for the user not to perceive a degradation in the service quality (e.g., jerkiness due to the erratic delivery of video frames). Some data loss may be tolerated but the data must be delivered within a bounded amount of time. This is considered to be the most difficult data type to transport due to its real-time requirements as well as its dynamic variability of bandwidth requirements [6, 20, 26, 28].

The type of video quality expected by today's users require very large network transmission speeds. Uncompressed speeds of broadcast-quality NTSC video and studio quality NTSC video are about 120 Mbits/sec and 216 Mbits/sec, respectively [6, 14]. Fortunately, recent advances in compression techniques [6, 12, 27] make the transmission of these high quality video media feasible. These compression techniques take advantage of the characteristics (or limitations) of the human visual system, to achieve lossy, yet visually lossless, compressed images and video. The two main video compression standards are commonly known as Joint Photographic Experts Group (JPEG) [6, 27] and Moving Picture Experts Group (MPEG) [3, 6, 12, 19]. JPEG is a still-image compression standard which can produce visually lossless compression with ratios up to 10 to 1. The MPEG motion video standard was designed to support full motion video at compression ratios up to 200 to 1. Bandwidth requirements for MPEG-2 range from 4 to 15 megabit per second stream of $640 \times 480 \times 24$ pixels per frame (24 bit color), 30 frames per second digital NTSC-quality video [6]. Both JPEG and MPEG standards are described in Section 2.

Related Work. Previously, the lack of ATM experimental work has been due to the lack of readily available high speed network components. However, during the past couple years, several vendors have begun providing ATM equipment. Preliminary performance results of ATM networks have been reported in [8, 21, 13, 24, 29]. These reports include mainly throughput and

delay studies. The performance of an actual ATM network under actual ATM traffic sources, such as VBR video traffic was largely unknown until a recent preliminary study [26] emerged which presented end-to-end performance evaluation of digital coded video (JPEG, MPEG-1 and MPEG-2) over a local ATM network (Fore Systems ASX-100 switch).

The goals of our study were the following:

- We sought to measure the ability of an actual ATM local area network to support real-time variable bit sources, i.e., bursty periodic data. Real-time variable bit rate traffic tolerates a small degree of loss but strict timing requirements on frame delivery must be met. Moreover, frame sizes may vary greatly even between consecutive frames. Previous work on the performance of packet switch networks to support variable bit rate sources have used either analytical means or simulation models to predict performance [5, 9, 17]. However, accurate analytical models which capture the time-varying correlated nature of the stochastic process that model video streams are usually intractable [15]. It is also obvious that even in simulation models it is not possible to capture all aspects (or even most) of an actual distributed system. In a distributed system, there exist many components, hardware and software, whose complex interactions cannot be naturally captured or predicted by a fixed model.
- It is likely that video transmission will be one of the dominant media types involved in distributed multimedia systems. Most non-trivial video-based multimedia applications entail the network support of many multiplexed coded video streams. Given an ATM environment, with a known maximum achievable throughput and delay, how many typical coded video streams (e.g., MPEG-1, MPEG-2, JPEG) can be supported within reasonable loss and jitter bounds ?
- We sought to determine the effect of switch architecture on the performance of digital video transmission. The Fore Systems ASX-200 and Cisco A100 Hyperswitch switch architectures provided an interesting basis for comparison of ATM switches architectures. Both switches rely upon Time Division Multiplexed (TDM) shared buses as the connecting switch fabric between input and output ports. However, both implement very different buffering strategies. We will present the effect of these different buffering strategies in our experiments.
- The Transmission Control Protocol (TCP/IP) and User Datagram Protocol (UDP/IP) suite of protocols are widely used today, and are very likely, during the initial deployment of ATM networks, to remain the dominant suite of protocols used in local and wide area network computing. How well is each suited for transmitting multiplexed video streams ?

This paper is organized as follows. Section 2 discusses the JPEG and MPEG video compression standards as well as the characteristics of coded video traffic. Section 3 discusses ATM switch architectures. Section 4 presents the experimental results. Section 5 presents a discussion of network requirements for the support of video transmission as well as the conclusion.

2 Digital Coded Video

As mentioned previously, video transmission is extremely bandwidth intensive. Thus sophisticated compression algorithms have been developed which are able to provide high quality, constant image quality video. In this section, we describe the JPEG and MPEG standards and their traffic characteristics.

JPEG [27] is an international digital image compression standard for continuous-tone (multi-level) still images (grayscale and color). It defines a “baseline” lossy algorithm, with optional extensions for progressive and hierarchical coding.

During the compression process, the pixel values of an image are divided into 8×8 blocks. Each block is transformed through a discrete cosine transform (DCT) function. The DCT is a relative of the Fourier transform and similarly produces a frequency map of 8×8 components. In the next step a “quantization coefficient” (or quantization factor) divides each 8×8 block into 64 frequency components. This is the fundamental information losing step. The high frequency components are normally reduced much more than the lower frequency components. Most of the quantized high-frequency DCT coefficients do not need to be sent because they have nearly zero values. The remaining DCT coefficients are encoded using either Huffman or arithmetic coding. The top left graph of Figure 1 depicts a sequence of JPEG frames from a Starwars clip.

MPEG-1 [12] defines a bit stream for compressed video and audio optimized to fit into a data rate of about 1.5 Mbits/sec. MPEG video compression is based upon exploiting temporal redundancy (moving image compression) as well as spatial redundancy. Moving image compression techniques predict motion from frame to frame in the temporal direction, and then use DCT’s to organize the redundancy in the spatial dimension. An MPEG data stream usually consists of three types of coded frames. ‘I’ frames, or intra-frames, are coded as still images (similar to JPEG). They do not rely on information from any previous frames. ‘P’ frames, or predicted frames, are predicted from the most recently constructed ‘I’ or ‘P’ frame. ‘B’ frames, or bidirectional frames, are predicted from the closest two ‘I’ or ‘P’ frames, one from the past and one from the future. A typical sequence of MPEG frames looks like: IBBPBBPBBPBBPBBPIBBP..... ‘I’ frames are usually much larger than ‘B’ or ‘P’ frames. Figure 1 depicts a sequence of MPEG-1 frames from a Starwars clip. Since, there are ‘I’ frames every 16 frames, there are peaks which occur every 16 frames.

In MPEG, there are two parameters which may be varied by a user. The first is the quantization factor (similar to JPEG) the second is the interframe to intraframe ratio, i.e., the number of frames in a period divided by the number of ‘I’ frames. These parameters are of interest because they allow the user to specify the visual quality, as well as allow the user to indirectly specify the bit rate characteristics of the MPEG stream [19].

MPEG-2 is the second phase of MPEG. It deals with the high-quality coding of possibly interlaced video, of either standard or High Definition Television (HDTV). A wide range of applications, bit rates, resolutions, signal qualities and services are addressed, including all forms of digital storage media, television, broadcasting and communications. The MPEG-2 “main profile” baseline is intended to be suitable for use by the largest number of initial applications in terms of functionality and cost constraints. It supports bit rates of 2 to 15 Mbits/sec over cable, satellite and other broadcast channels. The basic compression techniques used in MPEG-2 are similar to those used in MPEG-1.

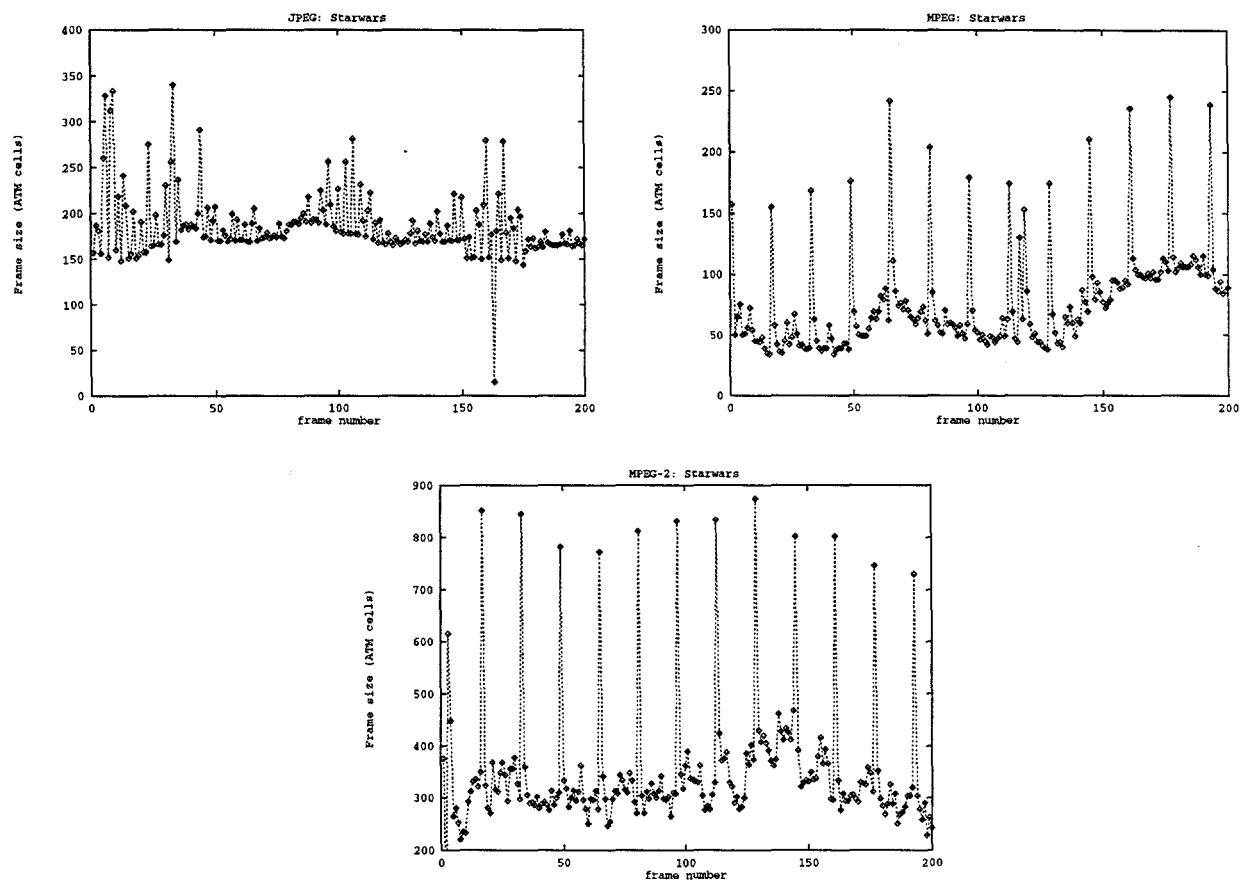


Figure 1: Top left graph: JPEG data stream. Top right graph: MPEG-1 data stream. Bottom graph: MPEG-2 data stream.

Video content. Another factor which influences the bit rate of a video sequence is the content of the video. High action scenes, scene changes, pans and zooms, reduce the amount of compression which can be used; the compression algorithm cannot rely on data redundancy. Video sequences with few scene changes, such as video-teleconferencing, generate relatively low constant bit rate streams. It has been observed that video data generated from a live teleconferencing application produces near constant JPEG data rates of about 1 Mbit/sec. Video sequences with complex spatial-temporal activity, such as sports sequences encoded using MPEG-2, require more than 10 Mbits/sec.

3 ATM Switch Architectures

In this section, we discuss several commonly used approaches to designing ATM cell switching architectures.

3.1 Space Division and Time Division Multiplexed Switches

Switches may be categorized as space division switches or time division multiplexed (TDM) switches. The major architectures in use are derived from one or both of these models. In both cases, the function of a switch is to transport (route) traffic from an input port to an output port. ATM's small fixed size cells allow efficient hardware design of switch fabrics that perform the switching and routing. As such, the switch fabric design is the core of the switch.

Wide area cell switches are usually space division switches. Typically, wide area switches must route very large bandwidths among a large number of links. These switches play a vital role in connecting sites together so they must also be very reliable, i.e., contain redundant paths. Hence wide area switches are usually constructed via many parallel interconnection devices. The internal switching capacity of such an interconnection network may be very high (upto 10 Gigabits per second). A parallel interconnection network is required for switches which must move more than just a few gigabits per second; bus technologies are capable of moving only a few gigabits per second.

Local area cell switches tend to be much smaller than wide area switches. A local area switch typically may support between 8 to 32 inputs and outputs, whereas a wide area switch may support thousands. The smaller size makes simpler technologies that scale relatively poorly, like busses and crossbars, more appealing. In a time division multiplexed (TDM) bus switch, the bus transfers cells from a source network interface (port) to one or more destination network interfaces (ports). Output buffering of cells may be provided by queues between the bus and ports, and by queues at each input or output port. As long as the capacity of the central switching fabric (bus) exceeds that of the individual ports, there is no contention at the ports or at the bus, and cells pass through the switch fabric with no delay (except possibly at the output buffers). This architecture is called "non-blocking". Since there is no port contention, provided that the bus speed exceeds the aggregate port speed, the performance bottleneck is the bus.

3.2 Buffering Strategies

A switch's buffering strategy may play at least as important a role in a switch's performance as the type of switching fabric employed. The tradeoff in buffer design is latency vs. throughput. Buffers are necessary to hold cells when there is contention at an input or output port. Holding cells in a buffer will increase delay, however, not buffering cells may decrease throughput.

Switches may contain internal and external buffers. Internal buffers usually occur as small buffers between switch elements in parallel interconnection (multistage) architectures. External buffers occur before and/or after cells pass through the switch fabric, i.e., external buffers exist as input or output buffers. The most common buffers are output port buffers.

Buffers may be organized as input, output, shared input, shared output, or shared input-output queues. Input queuing is the easiest to implement. However, performance may degrade substantially if the Head-Of-Line (HOL) blocking problem is allowed to occur. In the HOL blocking problem, a blocked cell at the head of an input queue prevents cells with unblocked paths from proceeding. The HOL blocking problem may be alleviated with special output queuing policies and operations such as windowing. Output queuing has been shown to be theoretically optimal. Shared output queuing has been shown to provide the optimal delay-throughput performance. On a port or connection basis, output buffers can be drawn from a common shared buffer pool, or may be completely non-shared in order to guarantee buffer availability to certain ports and connections.

Figure 2 depicts the two switch architectures of the ATM switches in our experimental testbed. These switches will be discussed in greater detail in the next two sections.

4 Experiments

4.1 Testbed Environment

The testbed environment consists of the following components:

- Eight IBM RISC System/6000 (RS/6000) workstations. Each RS/6000 has a 66 MHz POWER2 processor. The operating system used was AIX 3.2.5. Each RS/6000 was equipped with an IBM Turboways 100 ATM Adapter (TURBOWAYS adapter) [10]. The TURBOWAYS adapter uses an onboard 25 MHz Intel i960 processor. The adapter provides dedicated 100 Mbps full-duplex connectivity using PVCs, and Direct Memory Access (DMA) capabilities.
- A Fore Systems Forerunner ASX-200 ATM switch. The ASX-200 local ATM switch is Fore System's successor to the ASX-100 switch. Like the ASX-100, it is based on a 2.4 Gigabits per second switch fabric (TDM bus) and a RISC control processor. The Fore ASX-200 switch architecture provides output buffering but no input buffering (see Figure 2).

Its features include output buffers with a 13,312 cell capacity (per 2 output ports), dual leaky bucket policing, 'smart' buffers (per-VC queueing, multiple service priorities, dynamic buffer allocation), packet level discard and congestion control through the Explicit Forward Congestion Indicator.

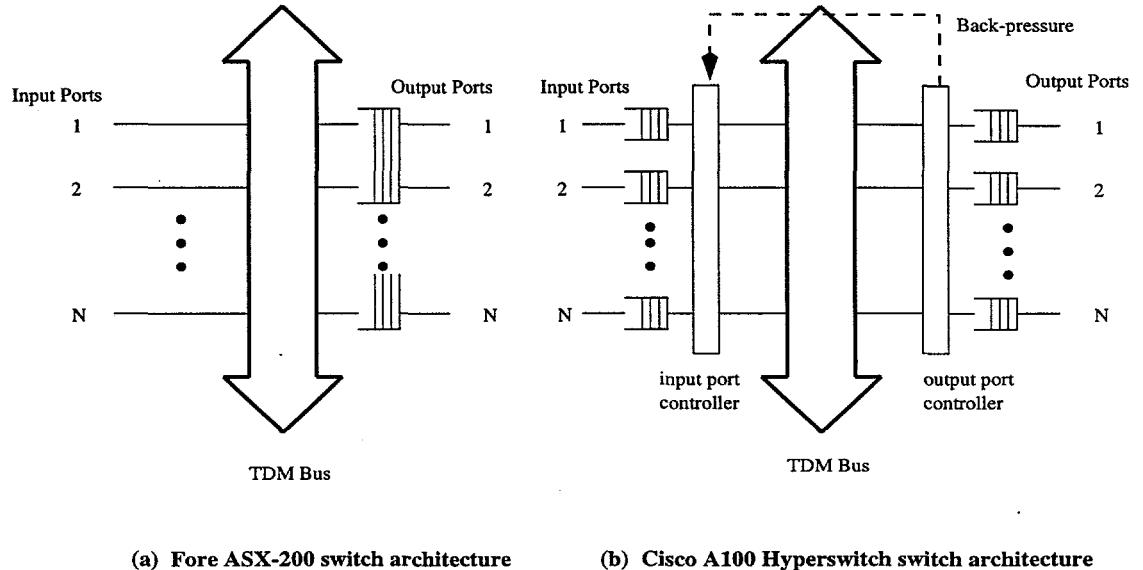


Figure 2: Switch Architectures

- A Cisco A100 Hyperswitch. The Hyperswitch is also based upon a 2.4 Gigabit per second TDM bus-based switch fabric. Its cells are routed through a combination of input and output buffers (see Figure 2). The input buffer size (per port) is 2450 cells and the output buffer size (per port) is 50 cells. Cells traverse through the switch fabric via a credit scheme which controls the admission of cells from the input buffers to the output buffers. In a credit-based schemes, cells are never admitted to a network component unless that component forwards a 'credit' [7]. In the Hyperswitch, cell loss never occurs at the output buffers because before a cell may be admitted to the output buffer, the output buffer must send a signal (or credit) to the input buffer to let it know that it has available buffer slot(s). The Hyperswitch's switch architecture was based upon the "Expandable ATOM switch (XATOM)" design proposed at NEC [7].

The Hyperswitch also provides support for two priority levels for both cell loss and cell delay. The switch offers full throughput for multicast and broadcast support.

- 100 Mbits/sec TAXI links interconnecting the RS/6000 hosts to the ATM switch. The topology of our experimental testbed consisted of an ATM switch (either the Fore or the Cisco) with direct TAXI connections to the 8 RS/6000 hosts. Connectivity between hosts was via one hop through the ATM switch.

The video data consisted of the following:

- **MPEG.** The input video stream for the MPEG codec was a 3 minute 40 second digitized from laser disc with a frame resolution (similar to NTSC broadcast quality) of 512×480

pixels. This particular Star Wars sequence was chosen because it contained a mix of high and low action scenes. The interframe to intraframe ratio was 16. The quantizer scale was 8. For these parameters, the image quality was judged to be good (constant) through the entire sequence of frames. The coded video was captured at 24 frames/second. The mean bit rate for the MPEG-1 sequence was approximately 1.5 Mbits/sec; the bit rate varied from about 0.3 Mbits/sec to 4.5 Mbits/sec. The mean bit rate for the MPEG-2 sequence was approximately 5.0 Mbits/sec; the bit rate varied from about 0.8 Mbits/sec to 10.5 Mbits/sec.

- **JPEG.** The input video stream for the JPEG codec was also a sequence from the Star Wars movie. The movie was input from a VCR and processed by a video card from Parallax Graphics, Inc., inside the Sparc 2. The frame resolution was 512×480 pixels. The quantization factor was 400. The image quality was judged to range from fair to marginal. The coded video was captured at 15 frames/second. The mean bit rate was approximately 2.3 Mbits/sec; the bit rate varied from about 2.0 Mbits/sec to 2.5 Mbits/sec.

It is important to note that the input sequence we chose contains high as well as low motion scenes. Hence the performance results cannot be compared to all types of video sequences. For instance, a video sequence generated from a video-conferencing application would contain, on average, smaller frames, since in most video-conferencing sequences there is little motion and few, if any, scene changes. Thus, better performance would be expected if such a sequence were used. We chose a clip with a high-level of motion in order to stress test the ATM network's ability to support bursty data.

We used the following configurations to measure how well the Fore ASX-200 switch and the Cisco Hyperswitch handled contention at the switch fabric, i.e., cross-traffic (**N senders to N receivers configuration**), and contention at the output ports (**N senders to 1 receiver**).

- **N senders to 1 receiver.** In this configuration, N RS/6000s simultaneously injected multiple video streams through the switch to a single receiver. See Figure 3.
- **N senders to N receivers.** In this configuration, N RS/6000s simultaneously injected multiple video streams through the switch to another disjoint set of N RS/6000s. Figure 4.

The following terminology and performance metrics are used throughout the remainder of the paper.

- The **average send time** of a frame is the time the first packet of a frame is transmitted to the time the first packet from the next consecutive frame is transmitted. When transmitting video frames at 24 frames per second, the average send time is $1/24 = 41.67$ milliseconds.
- The **received interval** is the time between the receipt of an entire frame (the last packet in a frame) and the receipt of the next entire frame. This metric is used to compute **jitter** - the interarrival delay between consecutive frames.

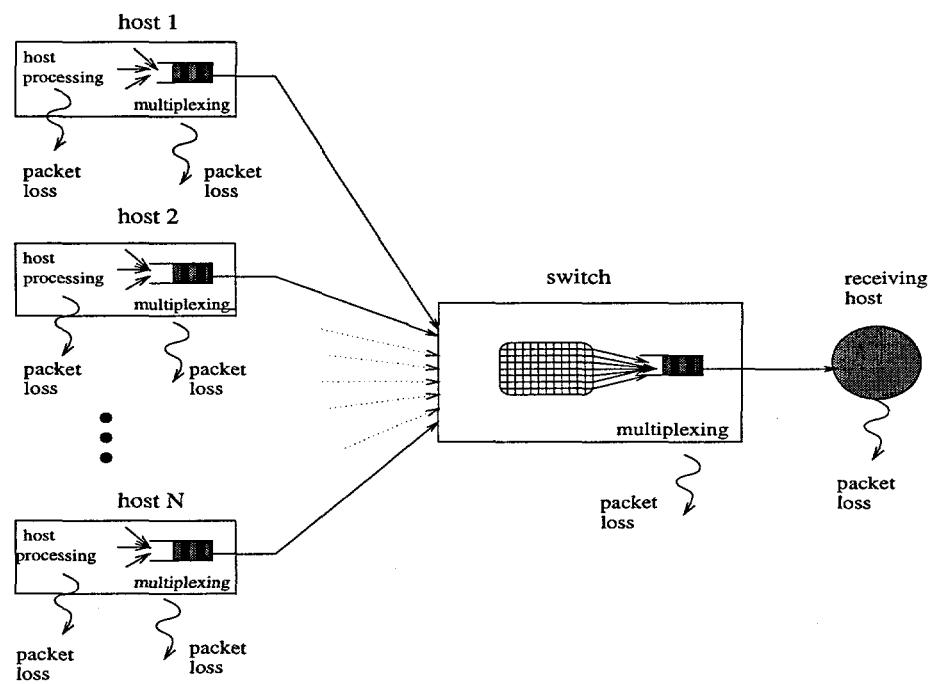


Figure 3: N senders to 1 receiver configuration

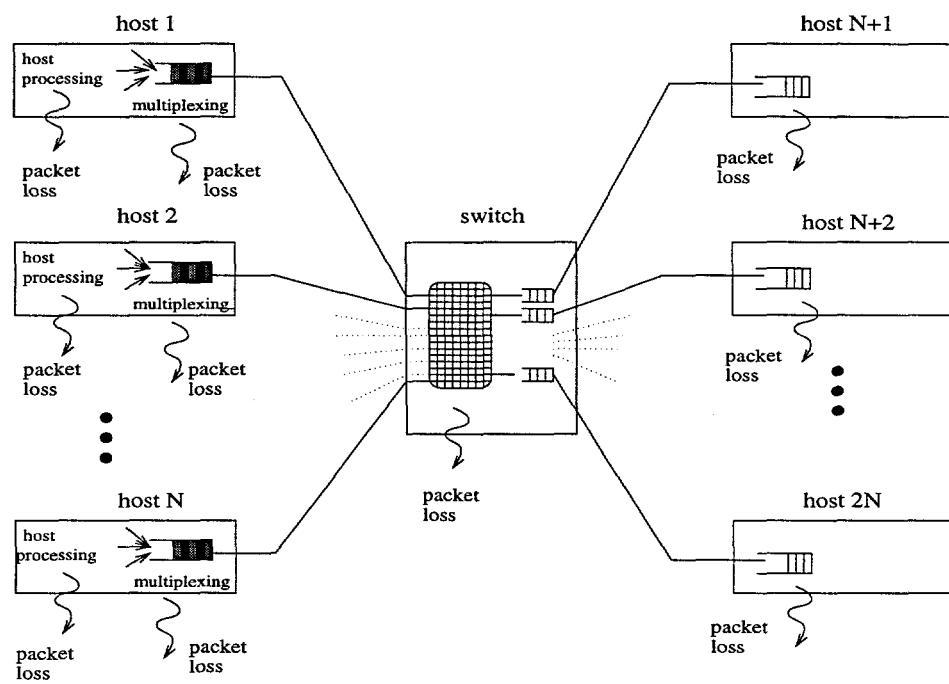


Figure 4: N senders to N receivers configuration

- The **fixed frame interval** is the interval based upon the frame rate, e.g. 24 frames per second implies a fixed frame interval of $1/24 = 41.67$ milliseconds. In a jitter-free environment, all of the received intervals would be equal to the fixed frame interval.
- The **percentage of lost frames** is the percentage of lost frames during a single run, or averaged among several single runs. Any frame missing a packet is considered lost.

Data Collection. In both configurations (Figures 3 and 4), the sender(s) and receiver maintained logs of packet departures and arrivals. The sending host recorded the number of packets sent, and the average send time. If the sending host was transmitting N streams (at 24 frames per second) and the average send time was larger than $1/24 = 42.67$ milliseconds, this situation implies that the sending host could not support the transmission of N video streams at the 24 frame per second rate, i.e., the sending host was overloaded, and hence was introducing jitter which was not created by the network components. In these cases, the data was not used in this study.

On the receiving side, the receiving host receives the packets and demultiplexes them according to sender host id and port number. A log for each incoming video stream is kept. The number of the missing frames is recorded. Any frame missing a packet is considered lost. The time to receive each entire frame is recorded. The time between the receipt of an entire frame and the receipt of the next entire frame is recorded.

For all our experiments, segments were ‘played’ for 3 minutes and 40 seconds. The data was ‘played’ by reading a file which consisted of an enumeration of frames and their corresponding byte sizes. To ensure accuracy, individual runs were executed multiple times. When appropriate, replicated results were averaged and the standard deviation computed and reported.

4.2 Previous Related Work

A previous study [16] has shown the Fore ASX-200 to outperform the Cisco Hyperswitch in terms of latency and jitter (see Figure 5). This was not a surprising result considering the additional complexity of the Cisco Hyperswitch’s buffering scheme. One purpose of this section is to see whether the additional functionality (credit-based input-output port buffering scheme), and hence overhead, as provided by the Cisco Hyperswitch, will serve to facilitate, or serve as an additional drawback, on the performance of video transmission. Comparing the Fore Systems ASX-200 with the Cisco Hyperswitch performance is particularly illuminating since the ASX-200 provides only output buffering.

4.3 Performance Measurements

Experiment #1: JPEG, MPEG-1, MPEG-2. In this experiment, the test configuration used was the 4 sender to 1 receiver model. The protocol suite used was UDP AAL5 ATM. The average (among all streams in the same run) percentage of frames lost as a function of the number of streams injected by the sending host is shown in Figure 6. JPEG, MPEG-1, and MPEG-2 sequences from the Star Wars movies are compared. Despite the poorer quality JPEG video sequence and the lower capture rate of 15 frames/sec (compared to 24 frames/sec for the

Test	Fore ASX-200	Cisco Hyperswitch A10
7.488 Mbps CBR stream switched across one port	10.57 microsecs	39.90 microsecs
7.488 Mbps CBR stream plus bursty IP data to 10% capacity switched across the same port	27.19 microsecs	52.67 microsecs
7.488 Mbps CBR stream plus burst IP data to 50% capacity switched across the same port	77.89 microsecs	130.21 microsecs

Figure 5: Data collected at the ENL[16]: Latency per port under various loads.

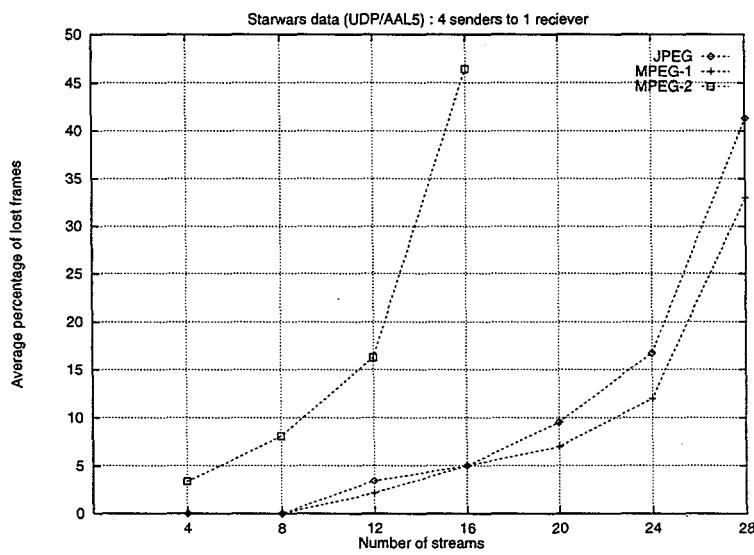


Figure 6: Experiment #1: Frame loss rates for JPEG, MPEG-1, and MPEG-2

MPEG-1 sequence), the percentage of lost frames for the JPEG coded sequence was still higher than the percentage of lost frames for the MPEG-1 coded sequence (see Figure 6). The higher loss rates of the JPEG streams can be attributed to the following. The average frame size in the JPEG coded sequence consisted of 10944 Bytes. The average frame size in the MPEG-1 coded sequence consisted of 5184 Bytes. The average 'I' frame size in the MPEG-1 sequence was close 11000 Bytes (which is close to the average frame size from the JPEG sequence). A sequence with an average larger frame size would result in greater losses because (i) any cell or packet loss would result in the entire frame being lost (i.e., smaller frames are less likely to be lost than larger frames), and (ii) a stream with a higher bit rate would cause more contention throughout its transmission and hence packets from that stream would be more likely to be discarded. For the same reason, due to the much larger frames in the MPEG-2 stream, the average percentage of lost frames for the MPEG-2 sequence was the greatest.

For the remainder of our experiments, we used the MPEG-1 sequence from the Star Wars movie. We chose to use the MPEG-1 sequence rather than the JPEG sequence because it is the most likely type of coded video to be used in the future; MPEG-1 provides higher compression ratios and lower mean bit rates than JPEG. The MPEG-2 stream data bit rate was too high (i.e., caused too many lost frames) to produce meaningful results in our particular environment. Thus, hereafter, the term MPEG implies MPEG-1.

As mentioned before, we used the TCP/IP and UDP/IP protocols as the transport mechanisms for transporting the video streams.

UDP and TCP. UDP provides connectionless datagram delivery service. It uses the underlying Internet Protocol (IP) to transport a message from one host to another host. It provides the same unreliable, connectionless datagram delivery semantics as IP. UDP does not maintain an end-to-end connection between the sending and receiving processes. It merely pushes the datagram out on the network and accepts incoming datagrams from the network. It does not provide guaranteed message delivery, nor in-order delivery, nor any type of flow control mechanism. The UDP layer is only responsible for multiplexing (demultiplexing) among multiple sources (destinations) within one host.

TCP is a connection-oriented protocol. It supports the reliable, sequenced and unduplicated flow of data without record boundaries. TCP supports guaranteed delivery (no loss) and flow control by using a sliding window protocol with time-outs and retransmits. TCP's greater capabilities, compared to UDP, is also its drawback: TCP requires more CPU processing and network bandwidth than UDP.

'Acceptable' visual quality. Continuous media traffic, such as coded video, have the real-time requirement that frames be displayed sequentially (continuously) with no prolonged delays between any pair of consecutive frames; the interarrival time, or jitter, between frames must be bounded.

When network congestion occurs, frames may be discarded (via buffer overflow), or in the case of TCP, may be discarded and then re-transmitted. If a frame arrives late, it will cause jerkiness in the visual medium. If a frame never arrives (is lost) its absence will also cause jerkiness in the visual medium. In MPEG, which consists of 'I', 'P', and 'B' frames, some

number of streams	Fore ASX-200				Cisco Hyperswitch			
	max	min	avg	std dev	max	min	avg	std dev
4 senders to 1 receiver								
32 streams (8 / sender)	94.10%	34.92%	72.25%	15.3	79.35%	2.32%	45.18%	11.0
28 streams (7 / sender)	75.02%	1.71%	32.71%	12.5	58.10%	2.49%	22.50%	9.1
24 streams (6 / streams)	33.83%	1.60%	11.81%	10.2	21.42%	1.46%	5.00%	2.6
20 streams (5 / sender)	24.80%	0.93%	6.74%	3.6	5.45%	0.32%	1.95%	0.8
16 streams (4 / sender)	9.35%	0.74%	5.32%	2.3	1.49%	0.26%	0.98%	0.5
2 senders to 1 receiver								
28 streams (14 / sender)	56.9%	1.42%	21.32%	11.6	42.9%	1.48%	18.21%	7.3
24 streams (12 / sender)	21.0%	1.40%	9.36%	5.6	14.1%	1.68%	4.63%	1.5
20 streams (10 / sender)	16.01%	0.73%	6.87%	4.8	3.6%	0.33%	1.99%	1.3
16 streams (8 / sender)	5.43%	0.25%	3.79%	1.6	1.12%	0.0%	1.02%	0.4

Figure 7: Experiment #2: N senders to 1 receiver: frame loss comparisons for different switches

frames are more important than others [12, 20]. ‘I’ frames are complete bit images. They must be received regularly in order to re-generate high quality images. ‘P’ and ‘B’ frames are used to ‘refresh’ the current image. If a ‘P’ or ‘B’ frame is lost, the decoder may be able to ‘guess’, or estimate, the lost frame until the next ‘I’ frame arrives. ‘I’ frames serve as a reference point for creating ‘B’ and ‘P’ frames. Hence, ‘I’ frames are more important to maintaining high visual quality; their loss is much more apparent to a viewer.

In our experiments, we define a loss of more than 10% to be visually noticeable to most viewers and hence unacceptable. The worst case occurs when all 10% of the lost frames are ‘I’ frames. At 24 frames per second, a 10% loss translates into a 21.6 frame per second frame rate. This is in the range of what is usually judged to be ‘acceptable’ visual quality. Recall that even the Parallax Graphics video card is only capable of displaying 15 frames per second.

Experiment #2: Output Port Contention Measurements In this experiment, the test configuration used was the N sender to 1 receiver model (see Figure 3). The protocol suite was UDP/AAL5/ATM.

Figure 7 depicts the 2 test scenarios. In the first scenario, we transmitted multiple video

number of streams	Fore ASX-200				Cisco Hyperswitch			
	max	min	avg	std dev	max	min	avg	std dev
128 streams (32 sender)	92.92 %	48.1%	73.46 %	12.1	76.91 %	2.19 %	40.25 %	14.2
112 streams (28 sender)	70.81 %	36.3%	53.32 %	9.9	68.89 %	3.14 %	23.82 %	11.3
96 streams (24 sender)	22.26%	1.8%	10.08%	5.0	23.32%	0.7%	9.69%	4.1
80 streams (20 sender)	13.16%	1.0%	2.55%	1.8	3.18%	0.4%	1.15%	0.4

Figure 8: Experiment #3: N senders to N receivers: frame loss comparisons for different switches

streams from **4 sending hosts to 1 receiving host**. In the second scenario, we transmitted multiple video streams from **2 sending hosts to 1 receiving host**.

This test was performed in order to measure how well each switch handled contention at an output port. Examining Figure 7 the first item to note is that given the same number of streams, for both the Fore and Cisco switch, the **2 senders to 1 receiver** case has a consistently lower frame loss rate than the **4 senders to 1 receiver** case. For instance, in the case where 28 streams are being transmitted to 1 receiver, the average frame loss rate for the **4 sender to 1 receiver** case was (32.71%, Fore switch) (22.5%, Cisco switch), and the average frame loss rate for the **2 sender to 1 receiver** case was (21.32%, Fore switch) (18.21%, Cisco switch). Given the same number of streams, while only varying the number of senders, implies that for a greater number of senders there is more contention at the output port and hence greater frame loss; for a fewer number of senders, more streams are already superimposed (multiplexed) at each sending host and less at the output port of the switch.

The comparison between how well the Cisco A100 Hyperswitch and Fore Systems ASX-200 switch handled output contention shows that the Cisco is able to consistently lose fewer frames than the Fore ASX-200 switch. Again, this may be attributed to there being less contention at the output port of the Cisco switch than the Fore switch. Recall, that the Cisco switch uses an input-output credit-based buffering scheme where loss never occurs at the output ports; data is only sent to an output port if that output port has sent a credit to the input port, where the data is being held, signalling that it has a certain amount of empty buffer space. Thus in the Cisco switch contention is distributed among the input ports whereas in the Fore switch (for the N sender to 1 receiver case) all the contention occurs at a single output port.

number of streams	Fore ASX-200			Cisco Hyperswitch		
	[2, 4)	[4, 6)	[6, infinity)	[2, 4)	[4, 6)	[6, infinity)
32 streams (8 / sender)	21%	13%	7%	17%	12%	5%
28 streams (7 / sender)	17%	6%	3%	14%	7%	2%
24 streams (6 / sender)	15%	4%	3%	12%	5%	0%
20 streams (5 / sender)	10%	3%	0%	6%	1%	0%

Figure 9: Experiment #4: 4 senders to 1 receiver - percentage delayed

Experiment #3: Cross Traffic Measurements. In this experiment, the test configuration used was the N sender to N receiver model (see Figure 4). The protocol suite was UDP/AAL5/ATM.

In this experiment, we transmitted multiple video streams between disjoint sets of 4 sending hosts and 4 receiving hosts, i.e., 4 pairs of sending-receiving hosts. Each pair of hosts supported the transmission of multiple MPEG streams. Figure 8 depicts the frame loss percentages as a function of load for both the Cisco and Fore switches. It can be seen that the switch fabric of the Cisco switch is able to handle the same traffic (as was handled by the Fore switch) but with a consistently lower percentage of lost frames. Again this may be attributed to the more sophisticated buffering scheme provided by the Cisco Hyperswitch. In the Cisco switch, contention is distributed among the input ports first whereas in the Fore switch all the contention occurs at the switch fabric (TDM bus) immediately.

Experiment #4: Jitter vs. number of multiplexed streams. This experiment was performed using TCP on ATM AAL5.

Figure 9 depicts the percentages of frames delayed for varying numbers of fixed frame intervals. Recall, that loss is not a performance measure in this experiment since TCP rarely loses packets. Packets which are delayed between x fixed frame intervals and $y - 1$ fixed frame intervals are said to be delayed by $[x, y)$, i.e., their received interval is between $[x, y)$.

Using TCP, packets which are discarded by the network are re-transmitted. Thus packets (and hence frames) are very rarely lost. Packets are discarded only after a pre-set number of re-transmissions occur. However, this situation occurs very rarely, and we did not observe any loss for any of the TCP experiments. We assumed that the receiving host could buffer up

to 3 fixed frame intervals (or 3 frames). This is a reasonable assumption because 'B' frames may reference either forward or backward frames; both the last and next 'P' or 'I' frames must be transmitted before the 'B' frames may be sent. Thus the receiver host must be able to buffer a minimal of 3 frames. Hence a frame received within a 3 fixed frame intervals will not cause jitter to occur. A frame which arrives outside of 3 fixed frame intervals is considered lost (undisplayable) and may contribute to the overall perceived jitter. Thus, similarly to what is acceptable for lost frames, we define an untolerable number of delays (i.e., losses) to be 10%; if more than 10% of the frames are delayed (beyond 3 fixed frame intervals), the visual quality is unacceptable.

Assuming frame delays less than or equal to 3 fixed frame intervals are tolerable (due to buffering), TCP can support upto (28, Fore switch) and (30, Cisco switch) multiplexed streams with only 9% of the received intervals exceeding 3 fixed frame intervals.

Note that TCP is able to support a far greater number of multiplexed streams (within reasonable inter-arrival delay constraints) than UDP (within reasonable frame loss rates). Also for TCP the variation in jitter between individual multiplexed streams from the same run was found to be negligible. Recall that for UDP the variation in frame loss rates between individual multiplexed streams from the same run was very high.

5 Discussion of Results and Conclusion

Recently many new distributed multimedia applications have been proposed. These new applications are based upon the transmission of digital information such as coded video, voice, high-resolution images, data, and graphics. For the network provider, the challenge lies in providing adequate network support for these various data types to operate on a communications infrastructure with a shared switching and transmission facility.

From our results, we conclude with the following points of discussion.

- **Fast packet switching?** Much emphasis has been placed on the need for fast packet switches as a means of enabling the high transport speeds of ATM [1, 4, 7, 20, 25]. This need for fast packet switches has brought about the philosophy that switches with the simplest, yet efficient, architectures are the most desireable. The *ideal* architecture which has been cited is one which provides either no input buffering (or an input buffering scheme which does not have the 'head-of-line' blocking problem) and buffers at each output port [1, 16, 20]. This appears to have been the philosophy behind the switch architecture of the Fore Systems ATM switches. And, indeed, the Fore Systems switch does provide lower jitter and delay than most other available ATM switches including the Cisco Hyperswitch. This was shown in [16].

How do a few microsecond jitter differences, or even a couple hundred microsecond jitter differences translate into the application Quality of Service (QoS) for video transmission, i.e., frame inter-arrival time of 30-40 milliseconds ? From the results in our experiments, we can infer that there is not a direct translation and (within reasonable bounds) the jitter (latency) differences and end-to-end performance of video need not even be related.

From our experimental results, as well as the results in [16], we reason that the Cisco switch performed superiorly because it uses a more sophisticated buffering scheme (at input and output ports) which uses a form of flow control to 'smooth' the traffic between the input and output ports [7]; fewer cells are lost due to buffer overflow. The Fore switch uses pure output buffering so all contention must be resolved at the output buffer - hence a greater number of buffer overflows may occur. So in the end, it did not matter that the Cisco switch incurred more latency (jitter) than the Fore switch!

- **Network Transport Protocols.** Due to the real-time nature of video traffic a guaranteed service protocol, or a best-effort service protocol with real-time scheduling at individual switch nodes, has usually been suggested as the most appropriate type of protocol for multimedia traffic such as video. Protocols which provide end-to-end flow control and re-transmissions, such as TCP, have been considered unsuitable because it has usually been thought that the delays caused would result in packets missing their bounded delay requirements and hence becoming meaningless (discarded). From our experiments, we observed TCP's sliding window protocol to have the effect of smoothing traffic burstiness, as well as still being able to deliver packets within their deadlines. Thus our results showed that the appropriate type of flow control may greatly improve application level performance significantly over a best-effort transport mechanism. In our particular environment TCP worked very well. This is not to suggest TCP as an appropriate Actually, in other environments (e.g., a wide-area network) it is very unlikely TCP would perform as well. The important point is that explicit flow control, either or a combination of link-link or end-end mechanisms, must be used to improve the performance of real-time traffic such as video. Best-effort delivery, in conjunction with real-time scheduling at individual switch nodes, which has been widely suggested as the most appropriate transport mechanism for real-time traffic such as video is unlikely to be adequate.
- **Traffic Control.** Traffic control for VBR video traffic is difficult to implement due to the real-time nature of the traffic. But, we believe it to be a necessary measure to increase network utilization and provide more stringent guarantees to continuous media traffic.

When the hosts and the network are stressed, the experimental data show that the burstiness of the variable bit rate coded video streams is a significant factor in the resulting performance degradation. We saw that controlling burstiness (through end-end flow control or switch buffering techniques) results in significantly less packet losses. This implies a fewer number of re-transmissions for transport protocols which guarantee reliability through re-transmissions, such as TCP, as well as for unreliable transport protocols such as UDP. Losses are significant when transmitting traffic types with large pre-specified (non-changeable) frame sizes like coded video. If any packet of a frame is lost, the entire frame must be discarded. A relatively small percentage of lost ATM cells may translate into a relatively large frame loss percentage in data traffic (such as coded video) which must use large frame sizes. Our results show that traffic control results in a significantly decreased frame loss rate while maintaining acceptable jitter and loss bounds.

Acknowledgements: The authors would like to thank Helen Chen and Chien Fang, from Sandia National Laboratories, Livermore, for the careful review of this paper.

References

- [1] Biagioli, E., Coope, E., Samsom, R., "Designing a Practical ATM LAN", *IEEE Network*, pages 32-39, March 1993.
- [2] Boudec, J., "The Asynchronous Transfer Mode: A Tutorial", *Computer Networks and ISDN Systems*, Vol. 24, pp. 279-309, 1992.
- [3] Chiang, T., Anastassiou, D., "Hierarchical Coding of Digital Television", *IEEE Communications Magazine*, May 1994.
- [4] Chlamtac, I., Franta, W., "Rationale, Directions and Issues Surrounding High Speed Networks", *Proceedings of the IEEE*, January 1990.
- [5] Cohen, D., Heyman, D., "A Simulation study of Video Teleconferencing Traffic in ATM Networks", *Proceedings of IEEE INFOCOM 1993*.
- [6] Cole, B., "The Technology Framework", *IEEE Spectrum*, March 1993.
- [7] Fan, R., Suzuki, H., Yamada, K., Matsuura, N., "Expandable ATM Switch Architecture (XATOM) for ATM LANs", *CISCO Systems Journal*, 1994.
- [8] Guha A., Pavan A., Liu, J., Steeves, T., "Supporting Real-Time and Multimedia Applications on the Mercuri ATM Testbed", *IEEE Journal on Selected Areas in Communications: Special Issue on ATM LANs*, May, 1995.
- [9] Heffes, H., Lucantoni, D., "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", *IEEE Journal on Selected Areas of Communication*, Vol. 4, No. 6, Sept. 1986.
- [10] "TURBOWAYS 100 ATM Adapter: User's Guide", IBM, First, March 1994.
- [11] Kawarasaki, M., and Jabbari, B., "B-ISDN Architecture and Protocol", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, pp. 1405-1415, Dec. 1991.
- [12] Le Gall, D., "MPEG: A Video Compression Standard for Multimedia Applications", *Communications of the ACM*, Vol. 34, No. 4, April 1991.
- [13] Lin, M., Hsieh, J., Du, D., Thomas, J., MacDonald, J., Distributed Network Computing Over Local ATM Networks, *IEEE Journal on Selected Areas in Communications: Special Issue on ATM LAN Implementation and Experiences with an Emerging Technology*, Vol 13, No. 4, May 1995.
- [14] Lyles, J., Swinehart, D., "The Emerging Gigabit Environment and the Role of Local ATM", *IEEE Communications Magazine*, April 1992.
- [15] Maglaris, B., Anastassiou, D., Sen, P., Karlsson, G., Robbins, J., "Performance Models of Statistical Multiplexing in Packet Video Communications", *IEEE Transactions on Communications*, Vol. 36, No. 7, July 1988.

- [16] Mandeville, R., "The ATM Stress Test: Which Switches Survived?", *Data Communications*, March, 1995.
- [17] Norros, I., Roberts, J., Simonian, A., Virtamo, J., "The Superposition of Variable Bit Rate Sources in an ATM Multiplexer", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991.
- [18] Ott, T., Tabatabai, A., Lakshman, T.V., "A Scheme for Smoothing Delay Sensitive Traffic Offered by ATM Networks", *IEEE Proceedings of INFOCOM 1992*.
- [19] Pancha, P., El Zarki, M., "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Communications Magazine*, May 1994.
- [20] Partridge, C., "Gigabit Networking", Addison-Wesley Professional Computing Series.
- [21] Pavan A., Guha, A., Liu, J., Midani, M. and Pugaczewski, J., "Experimental Evaluation of Real-Time Support on the *Mercuri* Wide Area ATM Testbed", *HTC Technical Report No. SST-R95-003*, April 1995.
- [22] Shroff, N., Schwartz, M., "Video Modeling within Networks using Deterministic Smoothing at the Source", *IEEE Proceedings of INFOCOM 1994*, pp. 342-349.
- [23] Skelly, P., Dixit, S., Schwartz, "A Histogram-Based Model for Video Traffic Behaviour in an ATM Network Node with an Application to Congestion Control", *IEEE Proceedings of INFOCOM 1992*.
- [24] Thekkath, C.A., Levy, H.M., Lazowska, E.D., "Efficient Support for Multicomputing on ATM Networks", Technical Report TR 93-04-03, Department of Computer Science and Engineering, University of Washington, April 1993.
- [25] Tolmie, D., "Gigabit Netowrking", *IEEE LTS*, May 1992.
- [26] Tsang, R., Pavan, A., Du, D., "High Speed ATM Network Support for Video-based Distributed Applications", Technical Report 95-069, Computer Science Department, University of Minnesota, Twin Cities, also to appear in: *ACM Multimedia Systems Journal*.
- [27] Wallace, G., "The JPEG Still Picture Compression Standard", *Communications of the ACM*, Vol. 34, No. 4, April 1991.
- [28] Wasem, O., Gross, A., Tlapa, G., "Forecasting Broadband Demand Between Geographic Areas", *IEEE Communications Magazine*, February 1995.
- [29] Wolman, A., Voelker, G., Chandramohan, A., Thekkath, C.A., "Latency Analysis of TCP on an ATM Network", Technical Report TR 93-03-03, Department of Computer Science and Engineering, University of Washington, March 1993.