A 622 Mb/s LAN/WAN Gateway and Experiences with Wide Area ATM Networking

Joseph B. Evans, Douglas Niehaus, David W. Petr, Victor S. Frost, Gary J. Minden, and Benjamin J. Ewy, University of Kansas

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. A gateway has been developed and implemented not only to support communications between an ATM LAN and WAN at 622 Mb/s, but also to provide a platform for conducting network control and traffic research. In addition, 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. This article 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.

Networks based on asynchronous transfer mode (ATM) are evolving rapidly. In 1992, when the Multidimensional Applications Gigabit Internetwork Consortium (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 services are primarily at DS3 (45 Mb/s). The MAGIC testbed provided early evidence of how ATM performed over WANs.

TerraVision (a real-time terrain visualization system) [1] 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. 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 began, and even with the rapid development of ATM technology are not commonly in use today.

To demonstrate that these rates can be obtained in an ATM local area network (LAN)/WAN internetwork, we designed, implemented, and tested a flexible OC-12c (622 Mb/s) gateway between a gigabit LAN and WAN. The target LAN was the AN2 [2], 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 the University of Kansas (KU). These have been connected to the MAGIC network using both OC-12c and OC-3c interfaces. Field programmable gate arrays (FPGAs) were used in the gateway design. When they were combined with the on-board R3000 line card processor, they enabled the implementation of 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.

While the gateway was under development, MAGIC used multiple OC-3 links for experimentation.

These early experiments, which used Transmission Control Protocol/Internet Protocol (TCP/IP) over ATM in a WAN, indicated the existence of performance bottlenecks. Before proceeding to the higher rates that would be enabled by the gateway, these bottlenecks needed to be eliminated. The cause of these problems, congestion, was identified, and traffic-shaping algorithms (pacing) were developed and implemented. With appropriate traffic control, full use of available link capacity was demonstrated.

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 of

This research is partially supported by the Advanced Research Projects Agency (ARPA) under contract F19628-92-C-0080, Digital Equipment Corporation, the Kansas Technology Enterprise Corporation, and Sprint.

allowing 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. The experience with MAGIC proved that the information transport capabilities of ATM were preserved in moving from an ATM LAN to a WAN. Furthermore, the success of the OC-12c gateway showed that these capabilities will scale to high speeds.

The following section will provide an overview of the OC-12c gateway, traffic pacing, and real-time traffic control at 622 Mb/s. Details on these topics are provided in [3–7]. Online information can be found at http://www.magic.net/KU.

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

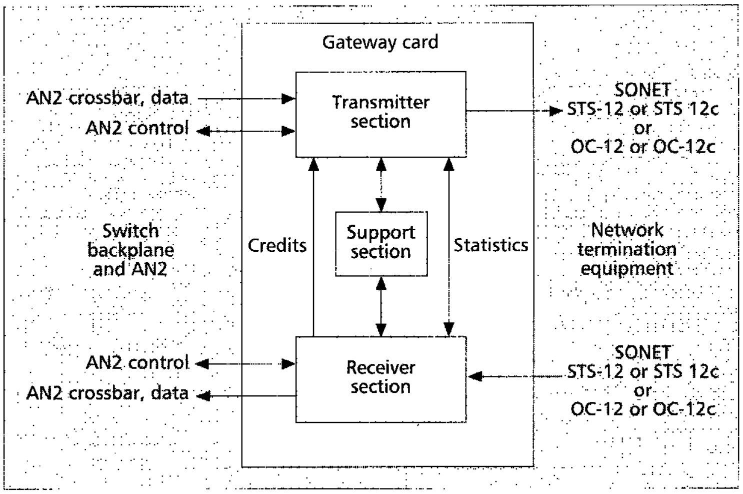
The use of similar technology in the LAN and WAN environments provides the opportunity for geographically distributed high-performance networks. A key element in realizing this goal is the development of efficient gateways, or user-network interfaces (UNIs), between the LAN and WAN environments. Although 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 to provide a high-level view of its implementation.

Gateway Overview

The target gigabit LAN in this work was the AN2 provided by Digital Equipment Corp. and developed by the DEC Systems Research Center [2]. The AN2 is a LAN based on ATM technology, consisting of switches 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 broadband integrated services digital network (B-ISDN) ATM traffic between the KU LAN and the MAGIC WAN at SONET OC-12 or OC-12c rates (622.08 Mb/s). The AN2/SONET gateway can support the following features:

- OC-12c interface to the WAN
- 4 x OC-3c streams contained in an OC-12 interface to the WAN
- 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, gateway support of TCP/IP traffic, but no restriction to that protocol suite
- Measurement of network performance

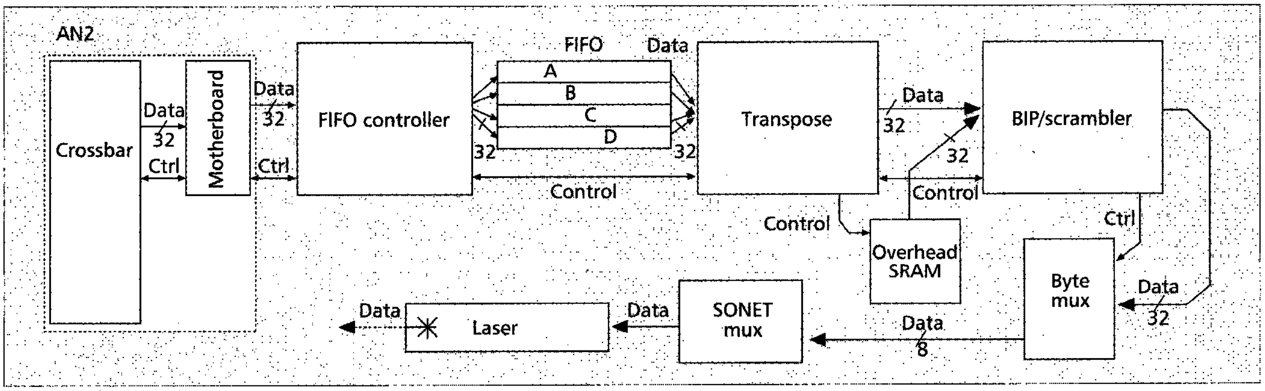


■ Figure 1. *AN2/SONET gateway.*

The AN2/SONET gateway is a single card that attaches to an AN2 motherboard, which then plugs into an AN2 switch port. The gateway, shown in Fig. 1, contains three primary subsystems: the receive section, the transmit section, and the support section.

Gateway Architecture

A block diagram of the transmit section of the gateway is shown in Fig. 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 the use of first-



■ Figure 2. *Block diagram of the transmit section of the gateway.*

in first-out (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 an FPGA in the transmit path, aptly named the FIFO controller. This chip resets the FIFOs, initializes them under the command of the 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.

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.

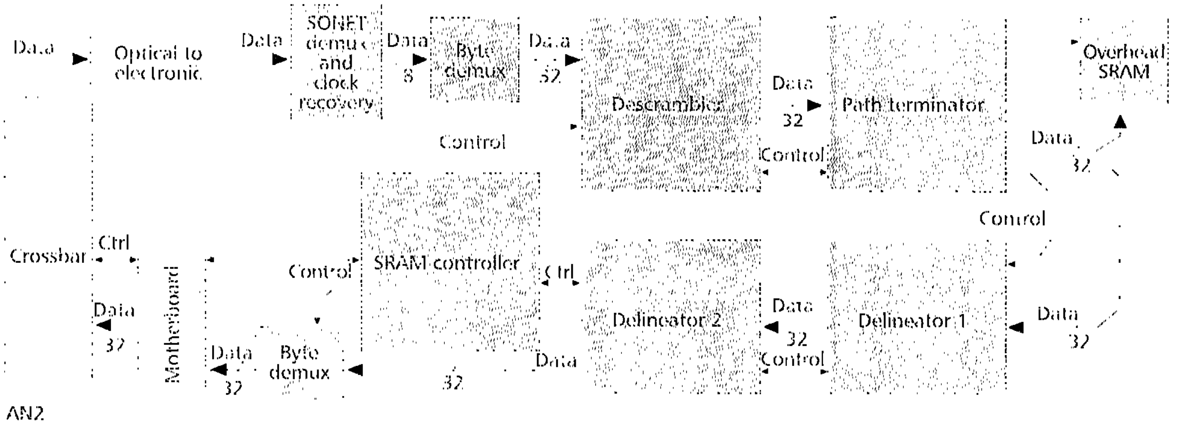
Most of the SONET section, line, and path overhead is stored in dual-ported serial random access memories (SRAMs). This enables the LCP to write in the fields that do not require high-speed processing in order to be computed. In order to simplify this entire process, pointers within the line overhead 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 data path. 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 means that most of the overhead for 16 frames can 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 segment or an integer number of byte-wide segments. The section and path BIP are 1 byte/SONET payload; hence they are 1 byte wide in the OC-12c mode and 4 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 pseudo noise (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 Fig. 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 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.

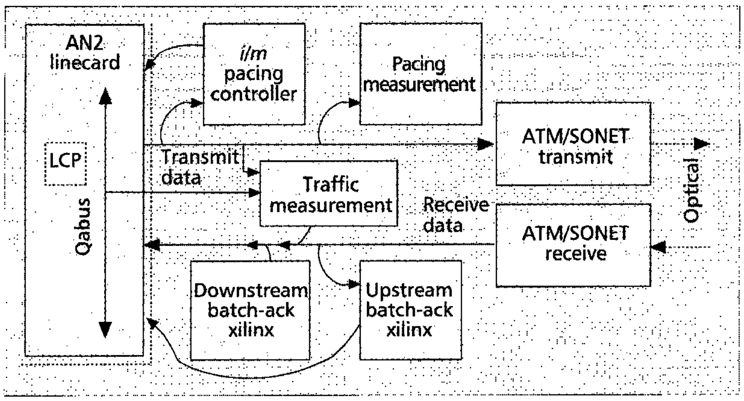
The second FPGA in the receive data path is the path terminator. This component determines the location of the start of the synchronous payload envelope (SPE) and the path overhead. It also provides the four least significant bits of address for the receive overhead SRAMs and the required control signals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



■ Figure 3. *Block diagram of the receive section of the gateway.*

header and verifying whether it matches the cyclic redundancy check (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.



■ Figure 4. *Block diagram of ATM support chips in gateway.*

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 Fig. 4.

The traffic measurement FPGA monitors the output and input port of the AN2 line card interface to the SONET gateway. The virtual channel identifiers (VCIs) as well as the ATM adaptation layer 5 (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 that 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 virtual circuit (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 independent 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 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 as described later.

The fourth and fifth support FPGAs implement the credit-based flow control scheme. The WAN flow control mechanism is implemented using two FPGAs plus several memories. The gateway supports the DEC FLOWmaster type flow control in which credits are transmitted in the cell headers, as well as an extension tailored for the WAN environment in which the credits are placed 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 video RAM (VRAM) on the motherboard. 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

Mode	Flow Control	Measurements
4 x OC-3c	Static i out of m rate control	VCI traces (VCI number vs. time slot)
OC-12c	Dynamic i-out-of-m rate control	AAL 5 traces (AAL 5 PDU start and stop times)
	WAN-based link-by-link credit flow control	

■ Table 1. *Summary of the features of the ATM/SONET gateway.*

Source — Fore ASX-200BX	Destination — DEC AN2	TCP Layer
SPARC 20	Alpha 3000/600	112.26
SPARC 20	Alpha 3000/600	124.27
Alpha 3000/400	Alpha 3000/600	101.73
Alpha 3000/400	Alpha 3000/400	122.40
Total		460.66

■ Table 2. *Gateway throughput.*

an interface that accepts the VCI and credit value. A summary of the features of the gateway is given in Table 1.

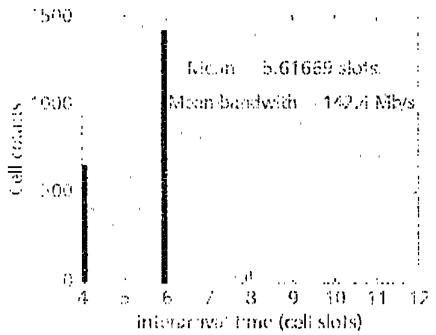
Gateway Results

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 & Operations Center (TIOC) in Overland Park, Kansas, and KU in Lawrence, Kansas. Two separate experiments were performed. Both tests involved DEC Alpha stations with OC-3c OTTO cards at each site. The tool for measuring application-layer throughput was ttcp.[1] The first experiment was performed without the use of flow control (pacing). Between 170 and 1200 packets/s were transmitted using a 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 arrival time series for this case is shown in Fig. 5.

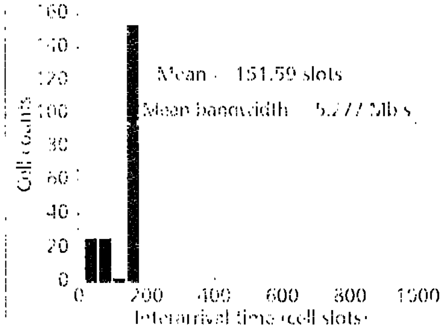
The second experiment was performed with flow control, that is, with cell-level pacing (cell-level pacing will be defined in the next section). In this case the pacing flow control was set to allow 5 Mb/s or 67 packets/s. Again, the chip was set up to sample the cell stream, and the observed mean cell level throughput was 5.3 Mb/s. There is some spread to the paced interarrival histogram (Fig. 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 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.

An experiment was conducted to evaluate the cell-level performance of the gateway. In this case, the ATM traffic was created by a set of four Pentium-based computers with an OC-3 interface running the Linux operating system. The Linux operating system was modified to have a personality that resembles a real-time system [9]. The operating system is responsible for scheduling the departure of packets, making it dedicated to the control of its ATM card and device driver. The end result is an inexpensive and reliable ATM traffic source.

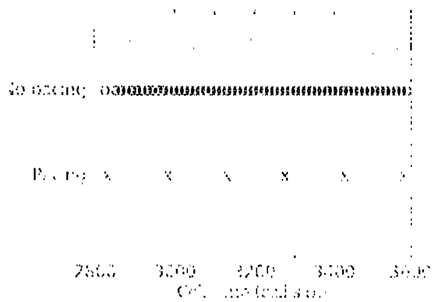
Cells were observed after the cell buffer SRAM shown



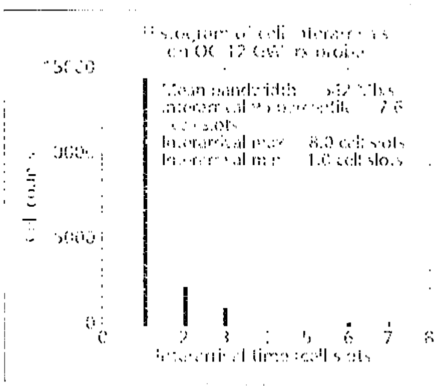
■ Figure 5. *Histogram of cell interarrivals for unpaced stream.*



■ Figure 6. *Histogram of cell interarrivals for paced stream.*

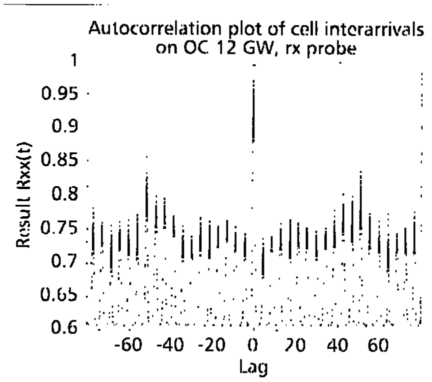


■ Figure 7. *Time series of cell arrivals for paced and unpaced streams.*

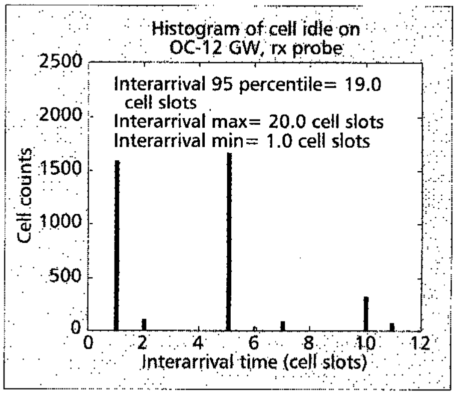


■ Figure 8. *Histogram of cell interarrivals of OC-12 GW, rx probe.*

[1] ttcp performs memory-to-memory copies using TCP over a network as fast as possible and measures the resulting throughput.



■ Figure 9. *Autocorrelation plot of cell interarrivals on OC-12 GW, rx probe.*



■ Figure 10. *Histogram of cell idle on OC-12 GW, rx probe.*

in Fig. 3. A histogram of the cell interarrivals is shown in Fig. 8. Note that the payload throughput observed was 542.6 Mb/s, or, including ATM and SONET overhead, over 600 Mb/s at the physical level. The correlation structure of this stream is shown in Fig. 9. Note the periodic nature of this flow, which can also be seen from the histogram of inter-idle-cell times shown in Fig. 10. A segment of the OC-12c cell time series is shown in Fig. 11.

To evaluate the TCP-layer performance as well as to demonstrate interoperability with other ATM switches (here a FORE ASX-200), an experiment based on the configuration shown in Fig. 12 was conducted.

Source traffic was generated using ttcp. The resulting TCP-layer performance is shown in Table 2. Note that only accounting for ATM overload results in an aggregate throughput of about 508 Mb/s.

There were no TCP packet errors observed in this experiment. In this case, the aggregate throughput is host-limited.

In addition to the above experiments, a pair of gateways, one in the Sprint TIOC and the other at KU (as shown in Fig. 13), were used to demonstrate interoperability with the SONET WAN equipment.

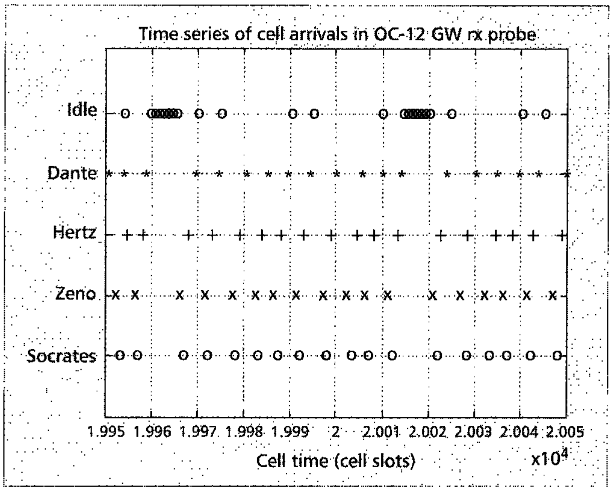
Experiences with Real-Time Traffic Control in ATM WANs

Understanding the performance characteristics of ATM WANs 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 poor 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.

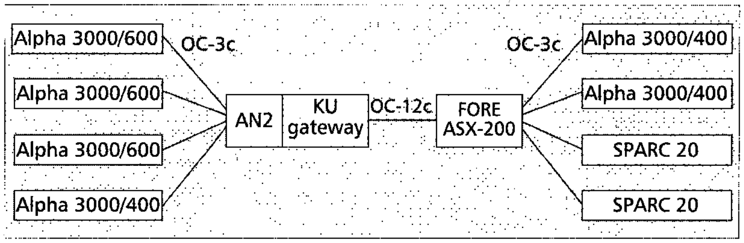
The Need for ATM-Layer Traffic Control in WANs

The initial measured performance of TCP/IP over the MAGIC SONET/ATM WAN was disappointing. An experiment was configured using a 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 to 100 Mb/s bandwidth constriction in the path. The experimental conditions in this case included 128-kbyte TCP windows and 64-kbyte ttcp write buffers. In this configuration, the throughput was only 870 kb/s, or less than 1 percent of the available capacity.

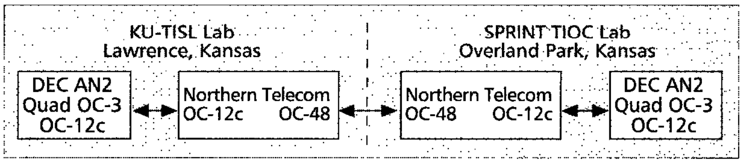
The performance with multiple traffic streams was also poor. In this case, two DEC 3000 AXPs (OC-3c) in Lawrence, Kansas, transmitted to a SPARCstation-10 (TAXI) in South Dakota to evaluate the effect of two hosts causing contention on the switch port to a third host. The experimental conditions included 128-kbyte TCP windows, 64-



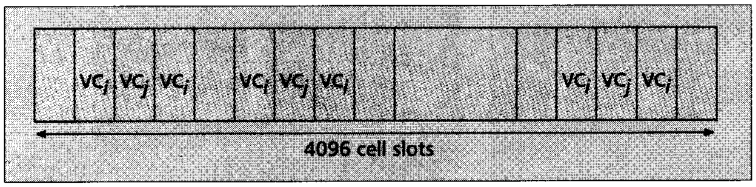
■ Figure 11. *Time series of cell arrivals in OC-12 GW, rx probe.*



■ Figure 12. *Gateway throughput configuration.*



■ Figure 13. *OC-12 WAN experiment.*



■ Figure 14. *OTTO pacing scheme.*

kbyte write buffers in ttcp, 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 retransmissions resulted in poor performance. Therefore, some form of ATM-level traffic control was needed.

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 as part of this effort 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 CBR traffic. Pacing can be enabled on a per-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 Fig. 14 will use up to half the available bandwidth if there is any traffic to send on that VC. The OTTO interfaces include a hardware implementation of AAL5, which was used exclusively in these tests, in hardware.

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 multiple streams in the cases just described. The rate went from 68.2 Mb/s to 52.3 Mb/s because the host was processing two streams instead of one.

Clearly, ATM-level pacing significantly improved the performance of the MAGIC network. Furthermore, these experiments increased the level of understanding of ATM WANs. See [6] for a complete description of the effects of traffic shaping on link utilization and congestion in ATM-based WANs.

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.

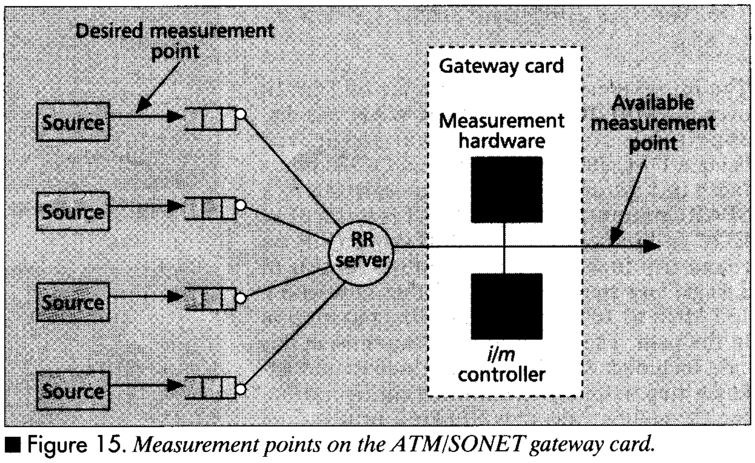
The basis of the algorithm developed here is the *i*/*m* rate control previously discussed. A user can transmit a maximum of *i* ATM cells in any *m* cell slot interval, yielding a maximum sustained throughput of *i*/*m* normalized to the line rate. The value of *i* also indicates the maximum number of consecutive cells which 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 Fig. 15. The traffic characteristics of the output of the round-robin queue differ from those of the source due to queuing 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 were found to 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 previously discussed.

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



■ Figure 15. *Measurement points on the ATM/SONET gateway card.*

cell at the beginning of a burst 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 16 shows a flow chart of the algorithm. The values for *i*, *m*, and the measurement interval need to initially be 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

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

This parameter measures how closely the controller is estimating the actual rate of the source. The algorithm begins by 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∗β then *i* should be increased, because the number of tokens assigned to the VC is too small. Similarly, if *i* is greater than 1.3∗β, then *i* should be decreased; 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. The parameters γ^+ and γ^- are the adaptive coefficients for rate.

The algorithm *m/i* actually adapts the inverse of the rate, so that the new value of m can be obtained via multiplication (*m/i* • *i*) rather than division ($\frac{i}{i/m}$).

Hence, the rate adaptation is performed according to the equations

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

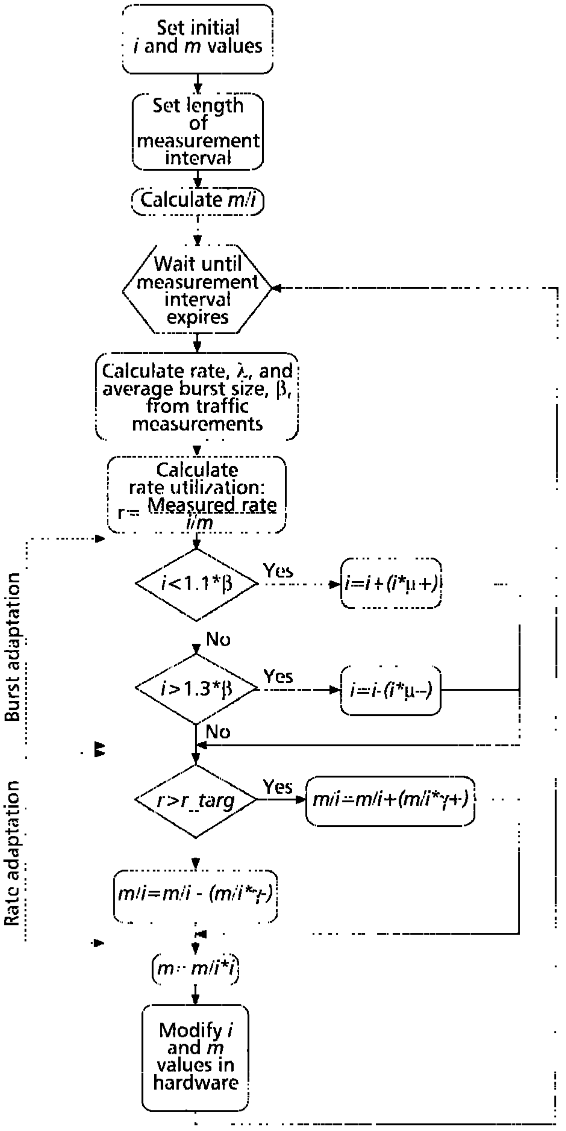
for an increase in the shaper rate, or

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

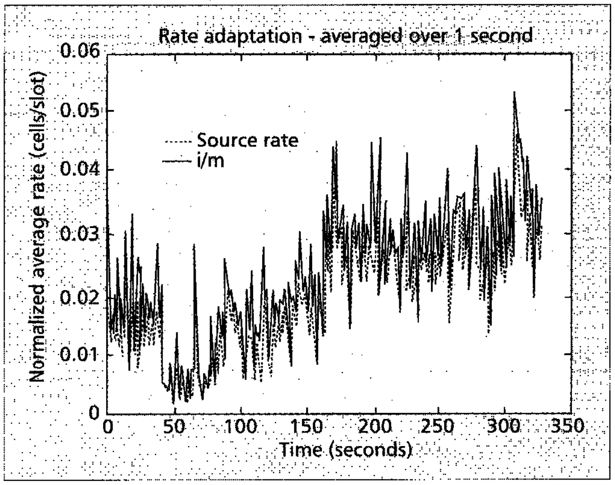
for a decrease in the shaper rate.

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 LCP indicated that a measurement interval in the range of 10 ms would be possible.

A simulation of the algorithm was performed [5] to determine its effectiveness in adapting to changes in measured inter-LAN traffic. The traffic trace used in this study


Set initial i and m values
Set length of measurement interval
Calculate m/i
Wait until measurement interval expires
Calculate rate, λ, and average burst size, β, from traffic measurements
Calculate rate utilization: r = Measured rate / i/m
Burst adaptation
i<1.1*β
Yes
i=i+(i*μ+)
No
i>1.3*β
Yes
i=i-(i*μ–)
No
Rate adaptation
r>r_targ
Yes
m/i=m/i+(m/i*γ+)
m/i=m/i - (m/i*γ-)
m=m/i*i
Modify i and m values in hardware

■ Figure 16. *Adaptive i-out-of-m shaping algorithm.*



■ Figure 17. *Rate adaptation of shaper to ethernet trace data.*

is inter-LAN Ethernet traffic obtained from Bell Communications Research (Bellcore) and represents a reasonable source that might submit traffic to a WAN.

Figure 17 shows the results obtained from the simulations using the trace data [8]. The data points on these figures were averaged over a 1-s interval. From Fig. 17 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. The shaper detects the large increase in the source rate, and in turn increases the assigned rate of that VC. Details of the implementation of this algorithm in the AN2 LAN/WAN gateway and additional performance results can be found in [5].

The adaptive i-out-of-m algorithm developed controls two system variables that translate directly into network policer parameters. By adaptively controlling the source average rate and maximum burst length, a smaller delay can be achieved. The 622 Mb/s hardware/software design of this algorithm demonstrates the feasibility of real-time control of high-speed flows.

Lessons Learned and Conclusions

Much was learned about the properties, performance, and limitations of a diverse internetwork containing gigabit ATM local area and wide area networks. First, it is possible to obtain high throughput at the TCP layer over TCP/IP-ATM WANs (e.g., over 120 Mb/s using OC-3c links). However, to achieve acceptable performance over such networks, appropriate flow control and proper end-systems configuration (e.g., large TCP windows) 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 WANs that contain rate mismatches and small switch buffers. Furthermore, cell-level pacing was experimentally demonstrated to be an effective scheme for coping with cell-level congestion in this high-speed WAN. Cell-level pacing is an open-loop mechanism. Closed-loop control, in this case hop-by-hop credit-based flow control, was also experimentally shown to be effective cell-level congestion management in the MAGIC network [7]. In uncontrolled LAN/WAN internetworks (i.e. networks without cell level control), it was also demonstrated that noncourteous flows (e.g., generated by UDP) can negatively affect the performance of other users. Clearly, cell-level control will be an important aspect of ATM WANs.

This effort also showed that OC-12c switch interfaces with advanced traffic control and measurement functions as well as adaptive cell-level traffic shaping at high rates are feasible. Such sophisticated dynamic bandwidth control mechanisms show promise but require careful implementations that take real-time programming issues into account. In the development of this hardware, it would have been useful to construct and test small prototypes of critical components, in this case the 622 Mb/s section of the gateways card, early in the development program. The use of FPGAs was critical to the success of this effort; furthermore, it should be noted that development with FPGAs should be treated as software projects.

Acknowledgments

Many graduate and undergraduate students contributed to the success of the KU MAGIC effort. We are grateful for the contributions of Hugo A. Uriona, Murat Bog, Cameron W. Braun, Darren Braun, Brian Buchanan, C. N. Gupta, Marcia Ramos, Ricardo J. Sanchez, Srini W. Seetharam, Vinai R. Sirkay, Michael T. Swink, and Esmaell Yousefi.

References

[1] Y. G. Leclerc and S. Q. Lau, Jr., "TerraVision: A Terrain Visualization System," SRI Int'l., Tech. Note #540, Menlo Park, CA, 1994.

[2] M. Goguen, "AN2: A Self-configuring Local ATM Network," *Proc. Nat'l. Commun. Forum (NCF)*, 1992.

[3] G. J. Minden et al., "An ATM WAN/LAN Gateway Architecture," *Proc. 2nd IEEE Symp. High Perf. Dist. Comp.*, July 1993.

[4] R. J. Sanchez, G. J. Minden, and J. B. Evans, "DISNES: A Distributed Network Emulation System for Gigabit Networks," *Proc. 1994 IEEE Comp. Aided Design and Modeling Workshop*, Apr. 1994.

[5] C. W. Braun *et al.*, "A High Speed Implementation of Adaptive Shaping for Dynamic Bandwidth Allocation," *Proc. 4th IEEE Symp. High Perf. Dist. Comp.*, Aug. 1995.

[6] B. J. Ewy *et al.*, "TCP/ATM Experiences in the MAGIC Testbed," *Proc. 4th IEEE Symp. High Perf. Dist. Comp.*, Aug. 1995.

[7] H. Zhu *et al.*, "Performance Evaluation of Congestion Control Mechanisms in ATM Networks," *Proc. 20th Int'l. Conf. Computer Measurement Group (CMG '95)*, Dec. 3–8, 1995, Nashville, TN.

[8] W. Leland *et al.*, "On the Self-Similar Nature of Ethernet Traffic," *Proc. ACM SIGCOMM*, 1993.

[9] D. Niehaus, http://www.tisl.ukans.edu/Projects/ATMRTS.

Biographies

JOSEPH B. EVANS [M] is currently an associate professor of electrical engineering and computer science at the University of Kansas. He received the B.S.E.E. degree from Lafayette College in 1983, and the M.S.E., M.A., and Ph.D. degrees from Princeton University in 1984, 1986, and 1989, respectively. His current research interests include high-speed (gigabit) networks as well as mobile and wireless systems. He is currently involved in research on the MAGIC, ACTS ATM Internetwork, and SPARTAN ATM WAN testbeds.

DOUGLAS NIEHAUS has been an assistant professor in the Electrical Engineering and Computer Science Department at the University of Kansas since 1993. His interests include real-time and distributed systems, operating systems, ATM networks, performance measurement, and programming environments. He received his Ph.D. in computer science from the University of Massachusetts at Amherst, where his thesis addressed the design and implementation of real-time systems. He was a system programmer at Bell Laboratories and AT&T Information Systems from 1981 to 1986, and at Convergent Technologies from 1986 to 1987.

DAVID W. PETR [SM] received his B.S.E.E. in 1976 from Southern Methodist University, M.S.E.E. in 1978 from Stanford University, and his Ph.D. in 1990 from the University of Kansas. He is an associate professor in the Electrical Engineering and Computer Science Department and the Telecommunications and Information Sciences Laboratory at the University of Kansas. His current research concerns the design and analysis of network resource management and traffic/congestion control mechanisms, particularly for integrated traffic ATM networks. Dr. Petr holds three patents from his work at AT&T Bell Laboratories between 1977 and 1986.

VICTOR S. FROST [SM] received the B.S., M.S., and Ph.D. degrees from the University of Kansas in 1977, 1978, and 1982, respectively. He is currently a professor of electrical engineering and computer science at KU. Dr. Frost has been the director of KU's Telecommunications and Information Sciences Laboratory since 1987. His current research interest is in the areas of integrated communication networks, high-speed networks, communications system analysis, and simulation. He is currently involved in research on the MAGIC, ACTS ATM Internetwork, and SPARTAN ATM WAN testbeds.

GARY J. MINDEN is an associate professor of electrical engineering and computer science at the University of Kansas and part of KU's Telecommunications and Information Sciences Laboratory. He earned the B.S.E.E. degree in 1973 and the Ph.D. degree in 1982, both from KU. In June 1994, he joined ARPA's Computer Systems Technology Office as a program manager in the area of high-performance networking technology. He will return to TISL in June 1996. Dr. Minden's research interests are in the areas of computer architecture and networks, machine intelligence, VLSI technology, and system models and simulation. Dr. Minden is a member of IEEE and ACM.

BENJAMIN J. EWY received the B.S.Co.E. degree in 1992 and the M.S.E.E. in 1994, both from the University of Kansas. Since 1992 he has worked as a research engineer at KU's Center for Research, Inc. During that time he has worked in the areas of gigabit WANs and wireless ATM networks. His current research interests include mobile computing and network security.