Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

A HIGH FIDELITY ATM TRAFFIC AND NETWORK SIMULATOR

Brian W. Unger
Fabian Gomes
Xiao Zhonge

Computer Science Department
University of Calgary
Calgary, Alberta T2N 1N4, CANADA

Pawel Gburzynski
Theodore Ono-Tesfaye
Srinivasan Ramaswamy

Computer Science Department
University of Alberta
Edmonton, Alberta T6G 2M7

Carey Williamson
Computer Science Department
University of Saskatchewan

Alan Covington
Jade Simulations International
Calgary, Alberta T2N 3P9

ABSTRACT

The design of an ATM Traffic and Network (ATM-TN) simulator which characterizes cell level network behavior is presented. The simulator incorporates three classes of ATM traffic source models: an aggregate ethernet model, an MPEG model and a World Wide Webb transactions model. Six classes of ATM switch architectures are modeled including output buffered, shared memory buffered and cross bar switch models, and then multistage switches which can be built from these three basic models.

The ATM-TN simulator can be used to characterize arbitrary ATM networks with dynamic multimedia traffic loads. Call set up and tear down via ATM signaling is implemented in addition to the various types of cell traffic streams generated by voice, video and data. The simulator is built on a simple, efficient simulation language called SimKit which is capable of supporting both fast sequential and parallel execution. Parallel execution is supported using WarpKit, an optimistically synchronized kernel that is aimed at shared memory multiprocessor platforms such as the Silicon Graphics Powerchallenge and Sun Sparc 1000 series machines.

The paper outlines general requirements for ATM traffic and network simulation, presents an ATM-TN simulator architecture, describes its major components and discusses the major issues associated with cell level ATM modeling and simulation.

1 INTRODUCTION

Asynchronous Transfer Mode (ATM) networks are rapidly emerging as the next generation communications standard. ATM offers substantial functional advances in communication services including high speed data transfer, and interactive voice and video. ATM multimedia technology is being adopted across a wide spectrum of network configurations including local area networks (LANs), metropolitan area networks (MANs), and wide area networks (WANs).

ATM can provide interactive multimedia communications to the desktop workstation. Desktop to desktop ATM architectures offer the greatest potential functional and performance gains. However, the large current investment in legacy networks such as ethernet, token ring, and frame relay require an evolutionary approach to the adoption of ATM. The current rapid development of ATM interfaces to legacy networks are supporting this transition to ATM across the spectrum of LAN, MAN and WAN configurations.

Despite this dramatic emergence of multimedia networking there are very significant technical problems that remain unresolved. These include issues in the definition of standards for call admission, congestion control and source policing, and research issues in scheduling, error control, switch design, bandwidth management, network management, multicasting, guaranteeing a quality of service (QOS) level and the performance of TCP/IP applications over ATM (Nikolaidis & Onvural 1992, Perloff & Reiss 1995, Vickers et al. 1993). Another area that must be addressed before ATM networks can become ubiquitous is accurate network planning tools. Network sizing tools which can generate candidate topologies and routings for a given set of projected traffic loads and for physical link and switch location constraints are needed.

Most of the above problems cannot be adequately addressed through analysis. The development of efficient, accurate analytic network design, planning and sizing tools, for example, requires high fidelity validated cell level simulation tools that can be used to validate these analysis tools. An understanding and analytic characterization of aggregated ATM traffic requires cell level simulation. The design of efficient applications that build on WWW and transport level protocols, as well as, the design and evaluation of enhanced transport level protocols, multicasting protocols, and network operation and management protocols requires accurate cell level simulation. The validation of cell level simulations against actual ATM network performance under a variety of traffic types and loads, and dynamic failure and congestion scenarios is also crucial.

This paper presents a cell level ATM traffic and network (ATM-TN) simulator that has been designed to address these issues. The ATM-TN development is part of a project called TeleSim. TeleSim is sponsored by an industrial consortium including: AGT Limited, Jade Simulations, Newbridge Networks, SAIC, Siemens ZFE, Silicon Graphics, the Stentor Resource Center and Canarie (the Canadian network for the advancement of research, industry and education). The TeleSim software development team includes faculty, research associates and graduate students at the Universities of Calgary, Alberta and Saskatchewan in Canada, and at Waikato University in New Zealand.

The ATM network protocols and design issues are not described in this paper. A recent collection of papers which provides an excellent overview of research issues in ATM networks can be found in ACM (1995).

This paper is organized as follows. First, the requirements that an ATM-TN simulator must meet to address current network planning problems and open research questions are outlined. Then the architecture of an ATM-TN that has the potential to meet these requirements is presented followed by brief descriptions of its major components. Finally, key modeling issues are discussed along with plans for future work.

2 ATM-TN SIMULATOR REQUIREMENTS

The general requirements of the ATM-TN simulator are to support: network performance analysis under varying traffic types and loads, network capacity planning, traffic aggregation studies, and ATM network protocol research. This spans a wide range of applications from production use by ATM network planners to ATM switch, network and protocol design by researchers.

These general requirements imply the need for cell level models of arbitrary network topologies with tens to hundreds of switches; models of multiple switch architectures and the characterization of a range of ATM network traffic sources. The requirements of the traffic models, switch and network models, simulator run-time performance, and other simulator requirements are outlined in the following paragraphs.

2.1 ATM Traffic Modeling

Three basic types of network traffic models are required: both compressed and uncompressed video traffic, an Internet traffic model of World Wide Webb (WWW) browsing interactions, and aggregate ethernet local area network (LAN) traffic. Specifically, each of the ATM-TN traffic models should support:

- point to point traffic definitions for multiple traffic types and varying traffic rates;
- dynamic bandwidth allocation and deallocation;
- statistics for cell loss ratios (CLR), cell transfer delay (CTD) and cell delay variation (CDV); and
- higher level traffic protocols and statistics specific to traffic types, e.g., TCP/IP packet level statistics such as error rates, packet sizes and delays.

All of the traffic models should capture the behavior of different types of user demands for communication services. Each type of traffic model will ultimately generate ATM cells as input to the network.

In the future, other traffic models in addition to the above will need to be incorporated. Modeling specific TCP/IP applications and LAN-ATM-LAN traffic including the (ATM Adaptation Layer) AAL functions are examples of anticipated future requirements.

2.2 ATM Switch & Network Modeling

The ATM-TN will model the protocols for ATM switching, call setup and release, and parts of the ATM Adaptation Layer (AAL). Other protocols such as policing and admission control will need to be modeled in the future. The latter will require the ATM-TN core models to be extensible.

The switch models must characterize the flow of ATM cells through the network as determined by the specific architecture of each switch type. The following outlines some of the elements that must be represented. Each type must be modeled using sub-components:

- a dimension (NxN) specifying the number of input ports and output ports on the switch;
- a switching fabric, defining the connections between input ports and output ports (e.g., crossbar, shared bus, banyan, delta network, ...);
- a set of buffers and a buffering strategy that specifies how many buffers are available, and how they are configured and used (e.g., shared vs. partitioned, input vs. output buffering); and
- routing tables, used to map cells from input ports to output ports. This mapping is done using virtual circuit and virtual path indices (VCIs and VPIs).

Links (or ports) must be modeled implicitly in the exchange of cells between switches (e.g., capacity in bits/sec, propagation delay, and bit error rate) rather than explicitly as a separate model component. This approach will significantly reduce the number of simulation events.

The intent is to make it easy to "plug and play" and evaluate the performance implications of different ATM switch architectures, and later, of different switch call admission, policing, and traffic control mechanisms.

2.3 Simulator Performance

The applications of the ATM-TN will require simulations of between 10^6 and 10^{12} cells. In a straightforward characterization of cell interactions through network switches this implies roughly five to ten times that number of simulation events.

Desirable simulation run-times are less than two hours with a maximum of 12 hours for overnight execution. These execution times are clearly not feasible for simulations of more than about 10^9 cells. Thus, extraordinary approaches will be required to achieve reasonable simulation run-times.

Initially it is not desirable to introduce approximations or hybrid analytic and simulation approaches. It is important that a high fidelity, accurate, model be developed first which can be used to validate simulator optimizations, and then approximate and hybrid approaches. Thus, despite the run-time requirements, an initial ATM-TN version is required that is capable of closely mimicking actual traffic and network behavior. Further, it is desirable that test and debug runs can be executed on ubiquitous PCs and Unix workstations.

In summary, the simulator efficiency will be crucial and care is needed to create very efficient sequential and parallel kernels.

2.4 Production versus Research Issues

A significant part of the ATM-TN simulator requirements are to support "production" ATM network sizing and planning by network planners. This requires a non-programmer interface that offers interactive data management and experiment control tools for large input and output files. The input files include an array of parameters to define specific switch and link characteristics, to define network configurations, and to specify experiment scenarios. Output files should be structured to enable the use of third party spreadsheet, statistical and graphical data analysis tools.

The other major goals of an ATM-TN simulator are to support research. The current ATM networking research issues of interest include: call admission control on entrance to the network, congestion control within the network, usage parameter control (UPC) within the network and switch service disciplines (scheduling policies). Other issues alluded to in the above sections include: multicast routing, switch architectures within network context and network management protocols.

Finally, there are research issues in modeling and simulation methodologies for this application. These include workload and traffic characterization in terms of measurable parameters, the design of efficient simulation kernels, hybrid analytic and simulation approaches, and the parallel execution of ATM traffic and network models. The requirements of an ATM-TN include being able to address these simulation methodology issues, the open issues in ATM networking outlined above, as well as, the commercially important network planning issues.

3 ATM-TN ARCHITECTURE

The ATM-TN simulator presented here was designed to meet all of the requirements outlined in section 2. The main design principles were to: (1) accurately mimic ATM network behavior at the cell level for specifiable traffic loads, (2) create a modular extensible architecture since requirements will evolve as research problems are solved and new issues emerge, (3) achieve reasonable execution times for ATM networks that consist of hundreds of nodes and tens to thousands of traffic sources. Since (1) implies an extremely computationally intense simulator and (2) implies a long lived simulator it was reasonable to expend a substantial effort aimed at also accomplishing (3). Further details on the ATM-TN can be found in Gburzynski (1995).

A cell level simulator for a moderately large ATM network has substantial potential for parallel execution. There is a great deal of independent activity involved in a large number of individual streams of cells flowing through a large network. Our preliminary analysis of the run-time behavior of an ATM-TN and related network simulation problems (Unger et al. 1994a, Unger & Xiao 1994b, Unger & Cleary 1993) suggests that optimistic synchronization schemes are relevant to this problem.

Although a decade of research in optimistic synchronization methods for achieving speedup through parallel execution on multiprocessor platforms has still produced mixed results, it was decided that the emergence of moderately large shared memory multiprocessor systems would have much greater potential for speedup in an ATM-TN simulator (Unger et al. 1993). This led to the design and development of SimKit and WarpKit (Gomes et al. 1995).

Simulations written in SimKit can be executed either sequentially or in parallel. The optimized sequential simulator (OSS) has been developed to support very fast, efficient, sequential execution. The second kernel, called WarpKit, supports parallel execution on shared memory multiprocessor platforms based on the Virtual Time (Time Warp) paradigm defined in Jefferson (1985) and Fujimoto (1990b).

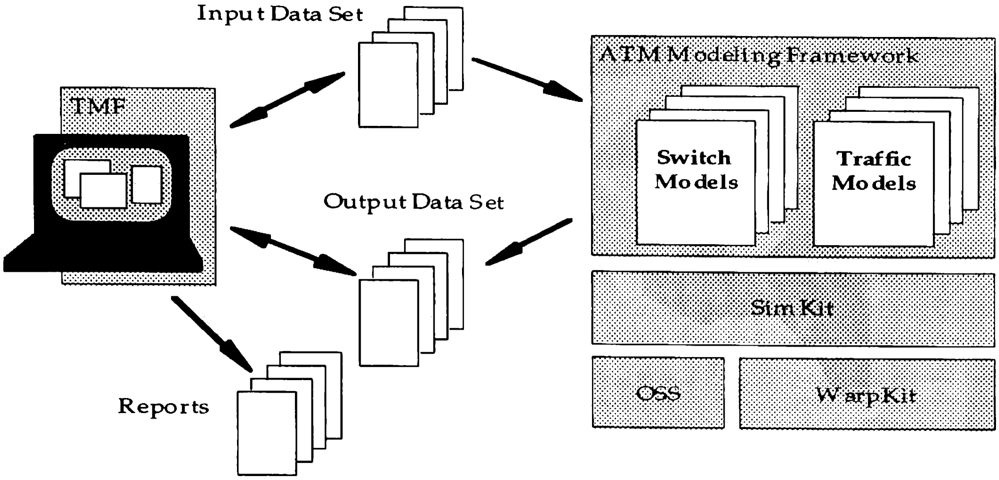

Figure 1: ATM-TN Simulator Architecture

The WarpKit kernel design is based on research reported in Fujimoto (1990a), Baezner et al. (1994) and Unger et al. (1990). The design of both WarpKit and SimKit are aimed at general purpose discrete event simulation problems. However, SimKit enables building custom mechanisms to support very efficient constructs that appear frequently in the implementation of network simulators, or specifically, in the ATM-TN.

The general structure of the ATM-TN simulator is illustrated in Figure 1. The major components include: traffic models, switch models, an ATM Modeling Framework, SimKit, WarpKit, OSS and the TeleCom Modeling Framework (TMF). The TMF and ATM Modeling Framework are outlined below. The traffic and switch models are described in subsequent sections.

The TMF enables the telecom engineer to configure a specific simulation model and vary model component parameters. The user constructs a model by specifying network topology, switch architectures, the traffic sources and the links connecting them. Network provisioning through Virtual Path assignments and Permanent Virtual Circuit allotments are possible. The input is saved in Unix files as per input data set specifications. Currently, testing the validity of input data, e.g., that values are within reasonable ranges, is performed by the ATM Modeling Framework (MF) at run time during the instantiation of various model components.

Simulation execution mode, the simulation executive, i.e., OSS or WarpKit, and the hardware platform for specific experiments are then selected through a menu driven run time control option in the TMF. The different types of reports to be generated from the simulation output files can also be directly selected through the TMF report generation menu option. Batch modes for simulation execution and report generation enable hands-free simulation execution requiring no subsequent interaction and monitoring during runs.

The ATM MF provides support and startup functions for the various model components. These include initialization and control, model construction and termination, model component identification, external event notification, and support for input and output. The MF also supports the modification of traffic and switch model components and the addition of new model components. Permanent and Switched Virtual Circuit modeling are supported using Virtual Path and Virtual Circuit connections. Generic Call Admission Control (CAC), and other Traffic Control functions are modeled in a modular way, supporting extension and user customization.

This architecture supports use of the ATM-TN simulator with or without the TMF user interface. Third party data management tools can be used to create and manipulate input data sets and the output data sets.

This architecture also supports exploring performance improvements along both sequential and parallel tracks. Once validation of the ATM-TN is complete we expect to explore a number of optimizations of the sequential simulator including mechanisms for cell/event aggregation and hybrid switch models. A great deal of work aimed at improving the parallel execution performance of the ATM-TN model is also possible.

4 TRAFFIC MODELS

Traffic sources are used to model any object that generates ATM cells that are to be carried by the network, i.e., by a network of switches. An ATM source can be an end-user application, a LAN, or the output port of another ATM switch.

Several distinct components are used in the construction of sources:

- *Input traffic sources*. These are used to model individual end-user applications that generate data carried by the network including data sources (e.g., FTP, Telnet, Mosaic, TCP/IP) and video sources (e.g., video-conferencing, JPEG, MPEG). Input traffic can modeled directly at the cell level, or at higher levels, e.g., burst level, packet level, message level, as appropriate for each source.
- *ATM Adaptation Layer* (AAL). This component is used to translate, where necessary, from input traffic source packets to ATM cells using the AAL specifications. This component is needed for converting packet or message level input traffic to cell-level models (e.g., for modeling TCP/IP, LAN-ATM-LAN traffic, and Frame Relay.) This piece is not needed for cell-level input traffic models.
- *Access control mechanisms*. This component will be used to model access control (policing) mechanisms, such as the Leaky Bucket, that may be interposed between a raw input traffic source and the ATM network itself, thereby changing the traffic flow characteristics. A person using the simulator can choose to run simulations either with or without this mechanism.

Each traffic type, contains a call generator that can spawn multiple instances of the traffic model at the site where the traffic type is attached. For the MPEG video model, call arrivals, call durations and call destinations are specified by the simulation user. WWW model arrivals are Poisson. The durations and destinations are specified by the WWW session model.

Many instances of MPEG and WWW transaction models may exist at a site, whereas only a single Ethernet traffic model (representing aggregate LAN traffic) can exist at a single site. The destination for a LAN traffic model is specified by the user. Ethernet packet sizes are not modeled explicitly.

Currently, the traffic descriptors for each traffic model are being determined. These will include the resource requirement parameters PCR, SCR, MBS, and the service requirement portion CDVT. Traffic sources keep track of the number of cells, packets and higher level data units transmitted and their corresponding retransmission counts. Call level statistics include the call blocking probability, the call setup / release delays and call durations. Traffic sinks maintain a count of cells, packets and higher data units received as well as discarded. Delay sensitive traffic models also maintain a per-cell CDV measure. Frame-level statistics are maintained by video sinks, which includes frames received and average number of frames lost or late.

The three basic traffic models, MPEG, ethernet and WWW, are outlined below. A detailed description of the traffic models can be found in Arlitt et al. (1995a). Further information on the WWW model is presented in Arlitt & Williamson (1995b) and on the ethernet model in Chen et al. (1995).

4.1 MPEG and JPEG Video Models

The MPEG/JPEG video traffic model is designed to characterize the unidirectional transmission of a single variable bit rate compressed video stream of data. This video data is both delay-sensitive and loss-sensitive; if cells are lost, image quality is degraded, and if cells arrive late, they are discarded, i.e., the effect of delay can be the same as if they were lost.

The MPEG (motion pictures expert group) algorithm is a standardized method of compressing full motion video for storage as digital data. An MPEG video stream consists of a sequence of images, or frames, that are displayed one after the other at short periodic intervals. The standard defines three types of three types of compressed frames called I, P and B frames as follows:

- *I frames* are encoded using only the image data available within the current frame. Thus an I (intraframe) frame represents a complete image, they provide an absolute reference point for the other two image types in an MPEG sequence.
- *P frames* contain motion-compensated data predicted from the preceding I or P frame. P (predicted) frames take longer to encode than I frames, are faster to decode than I frames, and achieve higher compression than I frames.
- *B frames* contain motion-compensated data from both the previous and the next frame (I or P). B (bi-directional interpolative) frames take the longest time to encode but offer the greatest compression.

A JPEG stream consists of all I frames and thus provides compression on each frame as an independent unit of data. MPEG sequences consist of a pattern of I,P and B frames called a group of pictures (GOP). The GOP frame is specified at the start of encoding, e.g., an IBBPBBPBBI sequence which is continually repeated.

The MPEG video traffic model in the ATM-TN simulates cell level ATM traffic generated by an MPEG video stream to a viewer. The model generates a given combination of I, P and B frames at a set frame rate. These random sequences are generated using the transform-expand-sample (TES) modeling methodology defined by Melamed and Hill (1995). Here, TES is used to generate three autocorrelated sequences of frame sizes, one for each type of MPEG frame. These separate sequences are then interleaved according to the GOP to produce the simulated traffic. Thus, this MPEG/JPEG traffic model is a composite TES model.

4.2 An Aggregate Ethernet Model

The ethernet LAN model is designed to represent aggregate data packet traffic on existing local area networks such as a university campus LAN. One of the requirements of the ATM-TN is to characterize ATM connections between legacy LAN networks as part of the evolution to ATM. This kind of traffic is likely to form a significant fraction of background load for early configurations of ATM networks.

Recent research confirms that ethernet traffic exhibits a fractal, self-similar, behavior where there is no natural length for bursts. This means that Poisson models do a very poor job of representing ethernet traffic. A stochastic process is said to be self-similar with Hurst parameter H if the process is covariance stationary and aggregate versions of the same process have the same structure as the original process. Thus, bursts are visible in aggregated traffic on a wide range of time scales, e.g., from milliseconds to minutes.

This self similar behavior has been observed both in traffic internal to an ethernet LAN, as well as, to traffic leaving a LAN. This behavior has been observed by researchers at Bellcore, and we have observed similar behavior in measurements taken at the University of Saskatchewan. The latter suggest self-similar behavior with a Hurst parameter close to 0.7. The ATM-TN aggregate ethernet traffic model is based on the TES methodology for generating random traffic sequences that have validated first and second order time series statistics, i.e., a frequency histogram and autocorrelation function that matches actual LAN measurement data.

4.3 A World Wide Web Transaction Model

The ATM-TN WWW model characterizes the cell traffic generated during a Mosaic like browser session exploring WWW servers. A single Mosaic session can generate one or more Mosaic conversations. Multiple destination sites may be involved in a single session, one destination at a time. Each conversation may consist of one or more TCP connections. Each TCP connection represents a single VC connection requiring an independent connection setup. A WWW session can spawn multiple conversations, e.g., from one to hundreds, and each conversation can spawn multiple connections, e.g., one to ten connections.

The input parameters to a WWW session include: mean number of conversations per session (Geometric distribution), mean conversation gap time (Poisson model), conversation destination, mean number of connections per conversation (geometric), mean connection gap times (Poisson) and the amount of information exchanged during a connection. In the latter, bytes sent by the source are modeled using a log Normal distribution, bytes received from the destination are modeled using a log Erlang distribution, and the mean and standard deviations are input parameters.

A single WWW session may last minutes or hours representing a user searching for information across the Internet. Empirical measurements of Internet WWW traffic collected at the University of Saskatchewan were used to construct and parameterize this model. Traces were collected using the Unix tcpdump and thousands of TCP connections were observed. Four one day traces were collected which contained 5,829 conversations which formed 57% of the total network activity.

5 SWITCH MODELS

The ATM-TN switch models currently include three simple, and three multistage switch models. These models support point-to-point switched virtual channels (SVCs) and permanent virtual channels (PVCs). The SVC call setup and release procedures modeled closely approximate the UNI 3.0 specification (ATM 1993). Virtual Paths (VPs) are also characterized and defined by input data as they tend to be long lived connections.

An ATM-TN network model has a number of components including: communication links, traffic source / sinks (TSSs), end nodes and switches. End nodes are a simplified type of switch that is used at the edges of an ATM network. The switch models provide:

- a simple traffic shaping scheme on a per VC basis;
- a simple call admission control (CAC) scheme at each switch;
- a simple user parameter control (UPC) scheme at each UNI (access switch);
- a network parameter control (NPC) scheme per NNI (internal switch);
- separate buffer queues for the different service categories (CBS/VBR/ABR/UBR) and a mechanism for scheduling the removal of cells from these queues. This is done at every output port; and
- separate thresholds for dropping cells with high loss priority and for forward loss congestion notification.

The basic operation of an ATM switch model is relatively simple. A cell that arrives on input port i with VCI=m and VPI=n is looked up in the routing table, and mapped to output port j with VCI=p and VPI=q. Real switches must be able to do this hundreds of thousands of times per second (e.g., 53-bytes @ 150 Mbps = 2.83 usec/cell). The VCI/VPI mappings are determined at the time of call setup, using an in-band signaling protocol.

The switch models also handle signaling for call setup and release which makes these models much more complex. VCIs are dynamic i.e., they are allocated and deallocated on a millisecond-second-minute time scale, while VPIs will be fairly static, i.e., they are allocated and deallocated on an hour-week-month time scale.

5.1 ATM Signaling

The model of signaling is one of the more complicated parts of the ATM-TN. However, signaling needs to be characterized since the overhead of end-to-end connection setup becomes significant as network speed increases. The model attempts to closely represent the UNI 3.0 specification (ATM 1993).

The model of network functions within switches follows the standard ATM layering, i.e., the network layer where the signaling protocol is implemented, the application layer at the end nodes which provide the interface to the traffic models, the signaling ATM adaptation layer which implements segmentation and reassembly, and the ATM layer which manages cell level transactions, are all characterized.

The signaling layer implements ATM connections, and thus the call setup and call release functions. Basic call admission and bandwidth allocation functions are represented. VP, VC, PVC and SVC connections are supported in the switch models.

5.2 Switch Model Architecture

The basic switch models have two major components, the control module and the switch fabric. When signaling cells arrive at a switch they are routed to the internal control module. The control module implements the signaling functions by acting on the information carried in these cells and sends out additional signaling cells of its own. This module also makes call admission and routing decisions, and updates the VPI/VCI translation tables within the switch.

The switch fabric transfers cells from an input port to the appropriate output port or to the control module. The switch buffering strategy is implemented in the fabric, e.g., output buffering, shared memory buffers, or a crossbar architecture. Multistage switches can be constructed from one of the basic switch types. Most larger switches contain banyon fabrics that can be scaled up to m any thousands of ports. A more complete description of the ATM-TN switch and signaling models is presented in Gburzynski et al. (1995).

6 CONCLUSIONS & FUTURE WORK

A prototype of the ATM-TN outlined above has been implemented and tested (Gurski et al. 1995). Preliminary experiments suggest that this simulator can be used to pursue research objectives in both ATM and simulation methodology, and can form the basis of production ATM network planning tools. Some of the specific research issues that we plan to explore in the future include:

- traffic control policies including: call admission and congestion control, usage parameter control (UPC), and switch service disciplines (scheduling policies);
- fast connection setup, e.g., by enabling the source to start sending its cells before the connection has been formally established and confirmed;
- models of aggregate multimedia traffic at various depths within an ATM network;
- dynamic bandwidth allocation in conjunction with call admission control;
- dynamic pricing strategies for bandwidth allocation and admission control;
- feedback based congestion control policies that may work with the Explicit Forward Congestion Indication (EFCI) mechanism;
- parallel simulation using optimistic synchronization on shared memory multiprocessor platforms; and
- hybrid models of ATM networks that include both analytic and "aggregate cell" model elements.

ACKNOWLEDGMENTS

The authors thank the TeleSim project sponsors including AGT Limited, Jade Simulations, Newbridge Networks, SAIC, Siemens ZFE, Silicon Graphics, the Stentor Resource Center and CANARIE (the Canadian Network for Advancement in Research, Industry and Education). The continued support of the Natural Sciences and Engineering Research Council of Canada is also acknowledged. Other significant contributors to this work include John Cleary and Richard Fujimoto.

REFERENCES

ACM. 1995. Special Edition on "Issues and Challenges in ATM Networks", *Communications of the ACM.*

Arlitt, M., Chen, Y., Gurski, R. and Williamson, C.L. 1995a. Traffic Modeling in the ATM-TN TeleSim Project: Design, Implementation, and Performance Evaluation, to appear in the *Proc. of the 1995 Summer Computer Simulation Conference*, Ottawa, Ontario, July.

Arlitt, M. and Williamson, C.L. 1995b. A Synthetic Workload Model for Internet Mosaic Traffic, to appear in the *Proc. of the 1995 Summer Computer Simulation Conference*, Ottawa, Ontario, July.

ATM Forum. 1993. ATM User-Network Interface Specification, Version 3.0, Prentice Hall, New Jersey.

Baezner, D., Lomow, G.A. and Unger, B.W. 1994. A Parallel Simulation Environment Based on Time Warp, *International Journal in Computer Simulation*, 4(2), 183-207.

Chen, Y., Deng, Z. and Williamson, C.L. 1995. A Model for Self-Similar Ethernet LAN Traffic : Design, Implementation and Performance Implications, to appear in the *Proc. of the 1995 Summer Computer Simulation Conference*, Ottawa, Ontario, July.

Fujimoto, R., 1990a. Time Warp on a Shared Memory Multiprocessor, *Transactions of the Society of Computer Simulation*, 6(3), 211-239.

Fujimoto, R.M. 1990b. Parallel Discrete Event Simulation, *Communications of the ACM*, 33(10), 30-53.

Gburzynski, P., Ono-Tesfaye, T. and Ramaswamy, S. 1995. Modeling ATM Networks in a Parallel Simulation Environment: A Case Study, to appear in the *Proc. of the 1995 Summer Computer Simulation Conference*, Ottawa, Ontario, July.

Gomes, F., Cleary, J., Covington, A., Franks., S., Unger, B., and Xiao, Z. 1995. SimKit: A High Performance Logical Process Simulation Class Library in C++, to appear in the *Proc. of the 1995 Winter Simulation Conference*, Washington D.C., December.

Gurski, R., Williamson, C., Arlitt, M., Covington, A., Gburzynski, P., Gomes, F., Ono-Tesfaye, T., Ramaswamy, S., Unger, B.W. and Xiao, Z. 1995. ATM-TN User Manual, WurcNet Technical Report, July, 12 pages.

Jefferson, D.R. 1985. Virtual Time, *ACM Transactions on Programming Languages and Systems*, 7(3), 404-425.

Melamed, B. and Hill, J. 1995. A Survey of TES Modeling Applications, *Simulation* 64 (6), 353 - 370, June.

Nikolaidis, I. and Onvural, R.O. 1992. "A Bibliography of Performance Issues in ATM Networks" Computer Communication Review, *ACM Sigcomm*, 22(5), 8-23.

Perloff, M. and Reiss, K. 1995. Improvements to TCP Performance in High-Speed ATM Networks, *Communications of the ACM*, 90-109, February.

Unger, B.W., Goetz D.J. and Maryka, S.W. 1994a. The Simulation of SS7 Common Channel Signaling, *IEEE Communications*, 32(3), 52-62.

Unger, B.W. and Xiao, Z. 1994b. The Fast Parallel Simulation of Telecommunication Networks, *Proc. of the 1994 Conerence on New Directions in Simulation for Manufacturing and Communications*, Operations Research Society of Japan, 9-16, August.

Unger, B.W., Arlitt, M., Gburzynski, P., Gomes, F., Gurski, R., Ono-Tesfaye, T., Ramaswamy, S., Williamson, C. and Xiao, Z. 1994c. ATM-TN System Design, WurcNet Inc. Technical Report, September, 142 pages.

Unger, B.W. and Cleary, J.G. 1993. Practical Parallel Discrete Event Simulation, *ORSA Journal on Computing,*, 5(3), 242-244.

Unger, B.W., Cleary, J. G., Dewar, A. and Xiao, Z. 1990. A Multi-Lingual Optimistic Distributed Simulator, *Transactions of the Society for Computer Simulation*, 7 (2), 121-152, June.

Vickers, B.J., Kim, J.B., Suda, T. and Hong, D.P. 1993. Congestion Control and Resource Management in Diverse ATM Environments, *IEICE Transactions on Communications*, November.

AUTHOR BIOGRAPHIES

BRIAN UNGER is a Professor of Computer Science at the University of Calgary and is President of WurcNet Inc. He leads the TeleSim Project. (http://www.ffa.ucalgary.ca/wnet/index.html)

ALAN COVINGTON is a member of the technical staff at Jade Simulations. He has worked for over ten years in the areas of parallel programming and simulation.

FABIAN GOMES is a Ph.D. student in the Computer Science Department at the University of Calgary. His thesis is on state-saving in optimistic simulation.

XIAO ZHONGE is a Research Associate of Brian Unger in Computer Science at the University of Calgary. He has worked extensively in parallel software development.

PAWEL GBURZYNSKI is an Associate Professor of Computer Science at the University of Alberta. His research interests include broadband network protocols, including ATM, traffic control policies and simulation.

THEODORE ONO-TESFAYE is a Ph.D. student in Computer Science at the University of Alberta. He is looking at issues related to fast ATM connection setup.

SRINIVASAN RAMASWAMY is a Ph.D. student in Computer Science at the University of Alberta. He is investigating dynamic bandwidth allocation to be used in conjunction with call admission control in ATM.

CAREY WILLIAMSON is an Assistant Professor of Computational Science at the University of Saskatchewan. His research interests include ATM traffic modeling and traffic control policies.