SimATM: An ATM Network Simulation Environment*

Antônio M. Alberti, Ernesto L. Andrade Neto, Jackson Klein, Leonardo S. Mendes
DECOM-FEEC/UNICAMP
P.O. Box 6101
13081-970 - Campinas - SP – BRAZIL
phone: +55-19-788-2053 fax: +55-19-239-1395
e-mail: alberti, ernestoa, jackson, lmendes@decom.fee.unicamp.br

Abstract – This paper describes the SimATM an environment to ATM network simulation. The SimATM is an application tailored to research, analysis and design of ATM networks developed in C++ to the Windows 95/NT™ operating system. The SimATM utilizes an event-driven simulation technique to achieve ATM simulation at the cell level. Simulation results are presented to demonstrate the SimATM simulation capabilities.

1. Introduction

ATM [1][2][3] is a connection oriented technology of packet commutation and multiplexing applied to the transport of small packets with fixed size, denominated cells, through a high-speed network. The ATM technology allows the integration and transport of voice, video and data traffics in the same network infrastructure. However, ATM is showing to be a very complex technology and still is under development. The cell switching permits the assembly of highly efficient high speed switches, resulting in high transmission rates, but the ATM network implementation needs an extremely complex infrastructure of overlaid protocols [4]. Because that ATM is a connection oriented technology that offers quality of service (QoS), it needs specific signaling and routing protocols, and a large sophisticated set of new mechanisms and traffic management functions. The development and performance evaluation of this new protocols and mechanisms required to ATM's technological evolution, as well as the design and analysis of ATM networks are very difficult tasks. Several approaches can be employed to address this problem: experiments with actual ATM network, experiments in ATM testbeds, and predictive techniques as mathematical analysis and simulation. Generally, some of these approaches have extremely high costs while others are applied to restrict cases. Among these, the employment of simulation tools represents a quite flexible solution. This paper describes the development and implementation of an ATM network simulation environment named SimATM. The SimATM aims to provide researchers and designers of networks with a tool for teaching, research, analysis and design of ATM networks. The SimATM was developed in C++ [5] to the Windows 95/NT™ operating system. The ATM simulation is performed at cell level and the simulation technique employed is event-driven.

The remaining of this paper is organized as follows. Section 2 describes the SimATM simulation environment. Section 3 presents the models developed for the simulator. Section 4 shows an example of ATM network simulation with CBR and VBR traffic sources. Section 5 shows simulation's results. Finally, section 6 draws the final conclusions, comments and directions for future work.

* The authors are supported by FAPESP.

2. SimATM Simulation Environment

The initial proposal to the SimATM development was to include the facilities needed to ATM network simulation in a communications systems simulator named SimNT [6][7]. The SimNT is a simulation environment for the development and analysis of communications systems and devices (especially optical communication systems), developed in C++ under the Windows 95/NT™ operating system. Inside SimNT a communication system is modeled as a set of interconnected blocks, which represent device models and numerical methods. The SimNT is a data-driven simulator. SimATM will include the same features of SimNT, like its graphic interface and expandable library of models, with a new kernel and commands to simulate ATM networks.

2.1 Modeling an ATM Network

The ATM network architecture modeled in the simulator is based on the B-ISDN Protocol Reference Model [8] (B-ISDN PRM). However, the SimATM ought to be an expansible tool, which allows the addition of new simulation facilities in a modular way. To achieve this goal the simulation framework was made to conform to a complete series of recommendations about functional operation of ATM equipment and management of ATM network elements defined by ITU-T (I.731 [9], I.732 [10] e I.751[11]). This framework permits the addition of new simulation facilities based on the functional architecture of the ATM network element.

Figure 1, shows the ATM simulation resources nowadays available in SimATM compared with the general architecture of an ATM network element defined in [9]. The present ATM simulation resources are restricted to transfer functions, layer management functions, and superior layers applications of the general functional architecture of an ATM network element. The implementation of these resources is done through layered models, the layers joined to compose the models of ATM network equipment and applications. The models of physical layer, ATM layer and ATM adaptation layer, partially include transfer functions and management functions of the corresponding B-ISDN PRM's layers. The remaining functions will be embodied in modular format in new simulator releases.

2.2 The SimATM Simulation Framework

The SimATM simulation framework was developed using an event-driven simulation technique [12] (Figure 2). The simulator's *kernel* has a command interpreter, an event queue, an event manager and several instances of ATM networks. The command interpreter executes commands over the several ATM networks. Only one ATM network can be simulated at once. The actual network components are modeled as ATM equipment or ATM application models composed of layer models and of queueing system models, which can schedule events in the event queue. The ATM equipment and applications are made of a common basic structure named block. The **block** denomination is applied to all simulator elements that inherit the block basic structure.

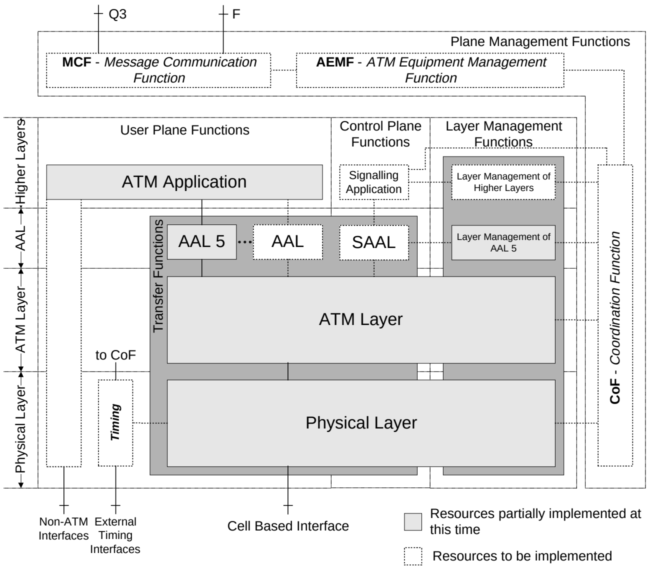

Figure 1: Present simulation resources of SimATM simulator.

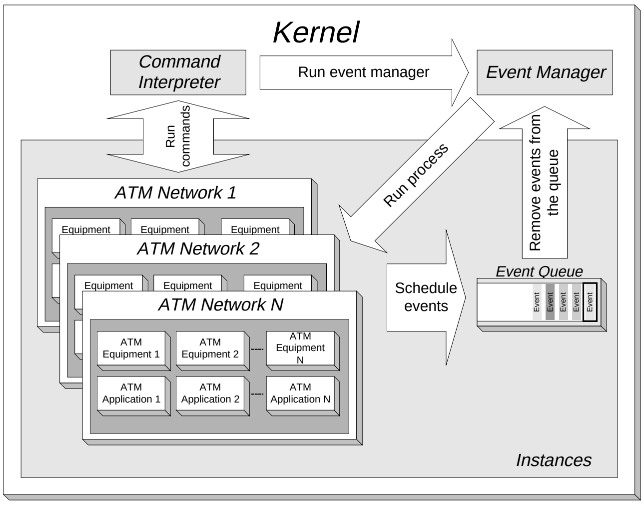

Figure 2: Present SimATM simulation framework.

The event queue holds the events that wait to be executed. The event manager takes away these events from the event queue and sends them to their destination equipment or application blocks inside the network under simulation. These blocks can follow events received from the event manager to the destination element's layer or its associated queueing system, then ending the event's life cycle in the simulator. The simulation continues until it reaches the predetermined maximum simulation time or until there are no more events left in the network's event queue to be executed. The behavior of layers and queueing systems models is described through process. A process in SimATM is defined as a set of tasks to be executed while event is defined as a message that fires process execution. When an event arrives to a specific layer or queueing system model, it fires the process that performs the actions specified in the event content.

2.2.1 ATM Network Structure

The structure of ATM networks inside SimATM is showed in Figure 3. The ATM network model has: one or more parameters, a data table, a parameter manager, a data table manager, a block manager, an ATM connection manager, a data connection manager, a neighborhood manager, an executor and several instances of ATM equipment and applications.

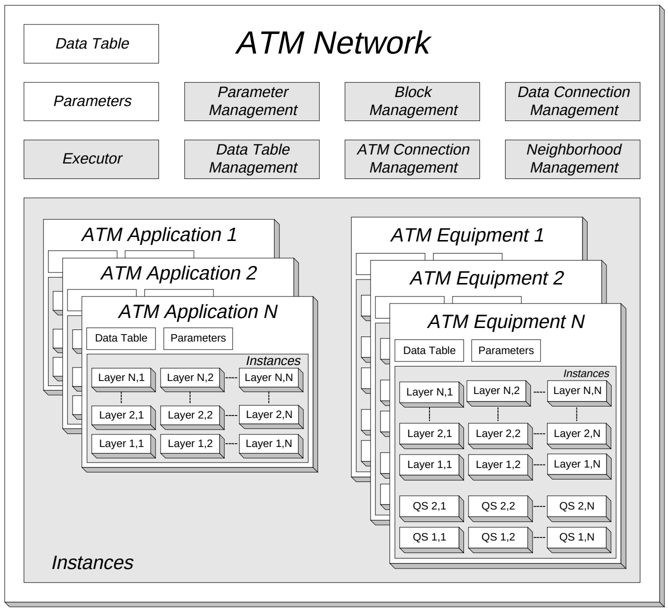

Figure 3: Structure of an ATM network inside SimATM.

The ATM equipment, has several instances of layer models and queueing system models. In its structure the ATM application has only instances of application's layer models, because there isn't cell storage in the application model. The equipment and application models also have one or more parameters and a data table. The parameter manager allows parameter handling inside ATM network blocks, as well inside block layers and block queueing system. The data table manager allows data table handling in ATM network blocks as well as in its layers. The block manager is in charge of creation, remotion, connection and disconnection of blocks from the ATM network. The ATM connection manager realizes the ATM network connections handling. The data connection manager does the data connections handling through the ATM network. These data connections are established over properly configured ATM connections and are used to send data through the network. The neighborhood manager performs the handling of block neighborhood information, assuring that the network blocks can be connected only after several conditions are satisfied. The executor initiate block layers and queueing systems before the simulation runs.

2.3 ATM Connections and Data Connections

In the SimATM there are two kinds of connections: ATM connections and data connections. The ATM connections are implemented as Permanent Virtual Connections (PVCs). The data connection uses the established ATM connections to transmit information between two ATM applications. If there isn't any established data connection no information will flow through the network.

The user needs to establish at least one ATM connection before the simulation execution startup. For each ATM connection in the network the simulator needs to keep in the network data table the VPI and VCI of each Virtual Channel Link (VCL). For the ATM connections the name of the ATM equipments that are part of the ATM connection are also kept. The network data table stores ATM connection's bandwidth and category of service (CBR, rt-VBR, nrt-VBR, ABR and UBR) and instant connection status (busy or available). For each data connection the network data table stores the connection category of service and bandwidth. It is also stored the names of the connection's ingress and egress ATM equipments, the names of the source and destiny applications, and the connection status.

3. Simulator Models

The SimATM has four kinds of models: queueing system models, layer models, equipment models and application models. These models will be further explained in the following sections.

3.1 Queueing System Model

The queueing system [13] model represents a client being served by a server. The clients wait in a queue to receive the service from one of the system servers. The SimATM queueing system model has only one server and one priority queue. The system clients, inside the model, are ATM cells waiting for the service in the queue (buffer). These cells are served one at once. The executed service varies according with the layer model that has the associated queueing system. For example, in the physical layer model the server executes cells transmission through a physical link. In the ATM layer model the server executes the remotion of cells from the switch output buffer. In both cases the service time is constant, however, it is possible the development of models with variable service time. The higher priority ATM cell stored in the queueing system buffer is removed first. The cell priority is calculated as a function of the cell CLP field and the category of service of cell connection.

Parameter	**Meaning**
BufferType	Identifies the buffer type to be used to store ATM cells (finite or infinite).
BufferSize	Buffer cell storage capacity.
TriggerByTimeStatus	Time sampling status.
TriggerByCellStatus	Event sampling status.
SampleStatisticsStatus	Statistics sample status.
LogEveryXSecs	Sets the time sampling interval.
LogEveryXCells	Sets the occurrence-sampling interval.

Table 1: SimATM queueing system model parameters.

The simulator queueing system parameters are show in Table 1. The queueing system model has a set of state variables that are based in the queueing theory [13]. These variables allow the complete determination of the queueing system state and are divided in two types: instantaneous and mean. These state variables can be logged to a file in two ways: by time or by occurrence.

In the time sampling all the variables are logged to a file in constant intervals of simulation time. In the occurrence sampling the state variables are logged to a file each time a cell arrives or leaves a queueing system.

			Sample by Time		**Sample by Occurrence**
	State Variable	**Variable Meaning**	File 1	File 2	File 3
Instantaneous Variables	Time	Time	x		x
	NoSRC	Number of System Received Cells	x		x
	NoBQC	Number of Buffer Queued Cells	x		x
	NoBDC	Number of Buffer Dropped Cells	x		x
	BS	Buffer Status	x		x
	NoBC	Number of Buffer Cells	x		x
	DoBC	Delay of Buffer Cells	x		x
	SS	Server Status	x		x
	NoSC	Number of Server Cells	x		x
	DoSC	Delay of Server Cells	x		x
Mean Variables	SMNoBC	Sample Mean NoBC		x	
	SVNoBC	Sample Variance NoBC		x	
	SMDoBC	Sample Mean DoBC		x	
	SVDoBC	Sample Variance DoBC		x	
	SMNoSC	Sample Mean NoSC		x	
	SVNoSC	Sample Variance NoSC		x	

Table 2: SimATM queueing system model state variables.

3.2 Layer Models

The layer models are derived from a basic structure named **layer**. Two kinds of layer models are developed: layer models for ATM equipment and layer models for ATM application. The layer models for ATM equipment are: ATM Adaptation Layer (AAL) type 5, Broadband Terminal Equipment (BTE) ATM layer, switch ATM layer and physical layer for the cell based interface. The layer models for ATM applications are CBR, VBR batch, and generic traffic sources and traffic receivers.

3.2.1 Equipment Layer Models

The equipment layer models are AAL model, ATM layer model and physical layer model. Only one protocol to the AAL is implemented, that for AAL 5. The other AALs models will be implemented in future simulator releases. The AAL 5 was chosen due to its widespread use and acceptation for ATM network data traffic. Some reasons for the widespread use of AAL 5 in ATM networks are presented in [3].

The ATM layer model was divided in two models: the BTE ATM layer model and the switch ATM layer model. For the physical layer only one model is implemented, the model of the physical layer for cell based interface.

The AAL 5 layer model is developed in conformance with

ITU-T recommendation I.363 [14]. The BTE and switch ATM layer models are developed in conformance with the recommendations I.361 [15], I.150 [16] and I.610 [17] of ITU-T. The model of the physical layer for cell based interface is developed in conformance with the recommendations I.432 [18] and I.610 [17] of ITU-T.

3.2.1.1 ATM Adaptation Layer Type 5 (AAL5)

This layer model constitutes the model for the hardware of the AAL 5 of B-ISDN PRM. The AAL 5 model has a null Service Specific Convergence Sublayer (SSCS). In this case the primitives exchanged with the layers above the AAL5 are the primitives of the Common Part Convergence Sublayer (CPCS). All the primitives that transport user information between AAL 5 and ATM layer and between AAL and the above layers are implemented. The AAL 5 has only one parameter that is the packetization delay of a cell (*CellPacketizationDelay*).

3.2.1.2 BTE ATM Layer

This model is a partial hardware model when compared with B-ISDN PRM. The ATM layer functions for data transmission and reception are executed. The BTE ATM layer model allows the insertion of OAM (Operation, Administration and Maintenance) cells, RM (Resource Management) cells and meta-signalling together with the user ATM cells flows. All the primitives exchanged in the user plane between ATM layer and physical layer and between ATM layer and AAL 5 are implemented.

The BTE's ATM layer model has the following parameters: *TriggerByCellStatus*, *SampleStatisticsStatus* and *LoggingEveryXSecs*. The parameter *TriggerByCellStatus* enables the log of cell waiting time across the network to a file. The parameter *SampleStatisticsStatus* enables or not the calculation and log of cell waiting time statistics in the network. When this parameter is enabled each time a cell arrives at the ATM layer of egress network equipment the cells waiting time statistics is updated. The parameter *LoggingEveryXSecs* configures the time interval between logs of the statistical variables to a log file.

3.2.1.3 Physical Layer for the Cell-Based Interface

This model constitutes the hardware model for the physical layer of the B-ISDN PRM. The instances of the physical layer model are connected with physical links (i.e., optical fiber, coaxial cable). The term **port** will be used to represent each instance of a physical layer connected through a physical link to another instance of the physical layer. Each port has one instance of the queueing system model associated with the physical layer. The functions to data transmission and reception at the physical layer are executed in the model. The physical layer model also executes the functions that allows the insertion of physical layer OAM cells (PLOAM) and the insertion of empty cells inside user ATM cells stream. These cells are generated with the frequency specified in recommendation I.432 of ITU-T.

The ATM cells received by the physical layer from ATM layer are stored in the queueing system associated with the physical layer and events are generated to follow each ATM cell to service or remove ATM cells that have already been served. In the physical layer model the service performed by the queueing system represents the transmission of a cell through one link connected to the next network equipment. The OAM and empty cells must be stored in the queueing system associated with the port, if needed. All the primitives changed in the user plane between ATM and physical layers are implemented.

The physical layer for cell based interface model has the following parameters: *BitRate* is transmission rate in bits per second; *SignalType*, is the description of the link rate; *Distance* that is the distance to the next ATM equipment in meters; *PropagationSpeed* is the signal propagation velocity in the transmission medium in meters per second; and *OAMTransmitionStatus* the status of transmission of OAM and empty cells.

3.2.1.4 Switch ATM Layer

This layer model constitutes the partial hardware model for the B-ISDN PRM's ATM layer. The switch ATM network model executes cell switching from a switch input port to a switch output port, passing through a Switch Fabric (SF) and through an output buffer. Each switch port has a queueing system associated with it. In this queueing system the ATM cells are stored. The model switch fabric is ideal, considering there are no cell loss inside the switch fabric and all cells are removed from the queueing systems server with a constant rate equal to the cell size in bits (424 *bits*) divided by the slot time. The slot time is the time allotted to remove a cell from a specified buffer. All the cells are followed to the respective output ports in parallel.

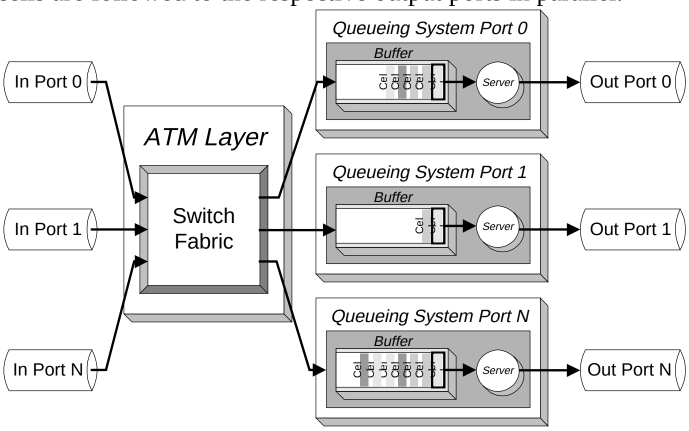

Figure 4: Switch ATM Layer Model

The primitives exchanged in the user plane between ATM and physical layer are implemented.

3.2.2 Application Layer Models

In the SimATM each traffic source or traffic receiver is implemented with a layer model for the ATM application. These traffic sources and receivers represent the upper levels of the B-ISDN PRM. The SimATM has models for the Constant Bit Rate (CBR) traffic source, Variable Bit Rate (VBR) batch traffic source, and generic traffic source and traffic receiver.

3.2.2.1 CBR Traffic Source

The CBR traffic model sends fixed size packets with a constant bit rate to the AAL of one ATM network equipment. The CBR traffic source continues its transmission until a maximum number of transmitted bytes are reached. It's also possible to specify the transmission starting time and the AAL to be used in the ATM equipment.

The CBR traffic source model has the following parameters: *BitRate* is the transmission rate in bits per second; *PacketLength* is the size of packet; *NumberOfBytesToBeSent* is the number of bytes to be transmitted; *StartTime* is the transmission starting time; *PacketName*; *AAL* is the AAL to be used; *ServiceCategorie* is the category of service to be employed; and *LogPacketTimeStatus* is the status of the packet's transmission time sampling to a log file. From the parameters *BitRate* and *PacketLength* it is defined the time interval between successive packet's transmission. The parameter *AAL* specifies the type of AAL that will be used in the equipment connect to the application. The parameter *ServiceCategorie* specifies the service category that should be used to transmit the source packets. The parameter *LogPacketTimeStatus* enables or disables sampling to a log file of the packets transmission time.

3.2.2.2 VBR Batch Traffic Source

The VBR batch traffic source model sends packets with mean size equal to the mean of a Poisson distribution at a mean time interval defined by the mean of a negative exponential distribution. It's possible to define the number of bytes to be transmitted, the transmission starting time, the AAL type to be employed in the receiving ATM equipment and the category of service of the ATM connection to be used.

The layer model of VBR batch traffic source has the following parameters. *NumberOfBytesToBeSent* is the number of bytes to be transmitted; *MeanNumberOfATMCells* is the mean packet size in ATM cells ATM; *MeanIntervalBetweenBatches* is the mean time interval between batches; *StartTime* is the transmission starting time; *PacketName* is the name of the packet; *AAL* is the type of AAL to be used; *ServiceCategorie* is the service category to be employed; *BitRate* is the transmission rate in bits per second; and *LogPacketTimeStatus* is the status of packet transmission time sampling.

3.2.2.3 Generic Traffic Source

The generic traffic source permits that the simulator can transmit packets which have transmission time and size read from a file. Then it is possible to simulate the effect of a real measured traffic pattern or another externally generated traffic model over the simulator network model.

The generic traffic source layer model has the following parameters: *BitRate* is the transmission rate in bits per second; *PacketName* name of the packet; *AAL* is the type of AAL to be used; *ServiceCategorie* is the service category to be employed; *SourceFileName* is the traffic source file to be sent through the network.

3.2.2.4 Traffic Receiver

This traffic receiver model is implemented as a layer model of the simulator ATM application. The traffic receiver gathers all the traffic destined to a network application. Each time that a packet arrives to its final destination it is possible to log the packet time statistics through the network.

3.3 Equipment Models

The SimATM equipment models are implemented over a basic structure called **ATM equipment**. Meanwhile, this structure is based in another basic structure called **block**. The SimATM equipment models are Broadband Terminal Equipment and ATM Switch.

3.3.1 Broadband Terminal Equipment (BTE)

The BTE model simulates a B-ISDN terminal equipment. This model is connected to one or more ATM applications and to other ATM equipment, which can be another BTE or ATM switch. The BTE model (Figure 5) has one instance of the AAL5, one instance of the BTE ATM layer model, and one instance of the physical layer for the cell based interface model with one instance of its associated queueing system model. The BTE model has the parameter number of applications (*NumberOfApplications*) that defines the maximum number of applications connected to the BTE.

The BTE model has also a data table that stores information related to ATM connections and data connections that pass through the BTE. In the data table are stored the ingress VPI and VCI of the ATM cells coming from each one of the BTE ATM connections and it also stores the ATM connections category of service.

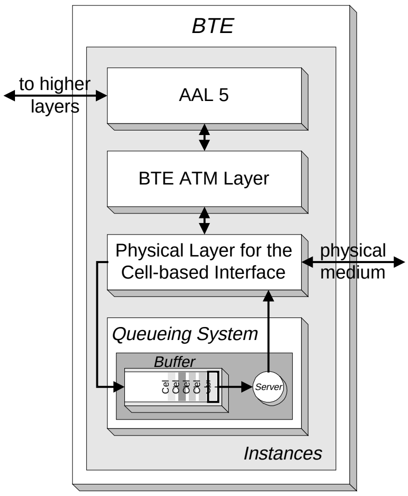

Figure 5: SimATM BTE model.

3.3.2 ATM Switch

The ATM switch model simulates a B-ISDN ATM switch. This model is connected to several BTEs or other switches. The switch model has an instance of the switch ATM layer model and several instances of the physical cell based layer model. For each connection with one BTE or other switch the switch model has one instance of the physical layer for the cell based interface model. The switch model also has several instances of the queueing systems associated with their respective physical layer for the cell based interface models. Figure 6 shows this switch model. The ATM switch model has the parameter

NumberOfPorts that defines the maximum number of ATM equipments that can be connected to the switch. The ATM switch model also has a data table that stores cell routing information: VPI and VCI of the input port and the input port name and VPI and VCI of the output port and the output port name. These information are stored to each established ATM connection.

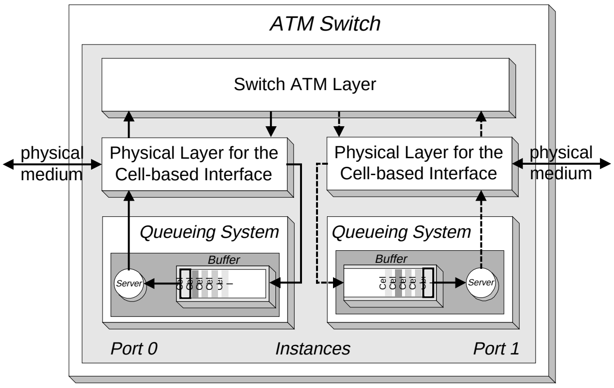

Figure 6: SimATM ATM switch model.

3.4 Application Models

The ATM application model comprehends the layer models to traffic generation and reception, which involves the interconnection functions and support to other applications. The SimATM has only one application model, the ATM Application. This model is implemented from a basic structure called **ATM application**. This structure is based in another basic structure named **block**.

3.4.1 ATM Application

The ATM application is the network traffic generator and receptor. The ATM application has the traffic source parameter (*TrafficSource*) that defines the kind of traffic source that will be used. These traffic sources are implemented as layers of the ATM application model.

The ATM application model has a data table that stores the category of service, the ATM connection name and bandwidth to be used by a established data connection of this application. Then it is possible that an application send packets through several data connections with distinct characteristics to other network applications, enabling ATM multicast simulation inside SimATM.

4. Simulation of an ATM network with CBR e VBR Batch Traffic Sources

The Figure 7 shows the configuration of the simulated ATM network. The application App_1 has one CBR traffic source, while the application App_2 has one VBR batch traffic source. The application App_3 also has one CBR traffic source, however no data is transmitted from this application that is used as traffic receiver in the simulation framework. Two ATM connections are established, one from BTE_1 to BTE_3, and the other from BTE_2 up to BTE_3. The CBR transmission is done with a data connection from App_1 to App_3, and VBR transmission with a data connection from App_2 to the same App_3. The CBR connection traffic has priority over the VBR connection traffic. These two data connections are established over the previously established ATM connections. The BTEs are connected through an ATM switch named SW_1.

The application App_1 transmits at a fixed rate of 18.56 Mbps packets with the size of 232 bytes. The application App_2 transmits packets with the mean size of 0.4 ATM cells and with a mean interval between packet batches of 2.7263 μsec.

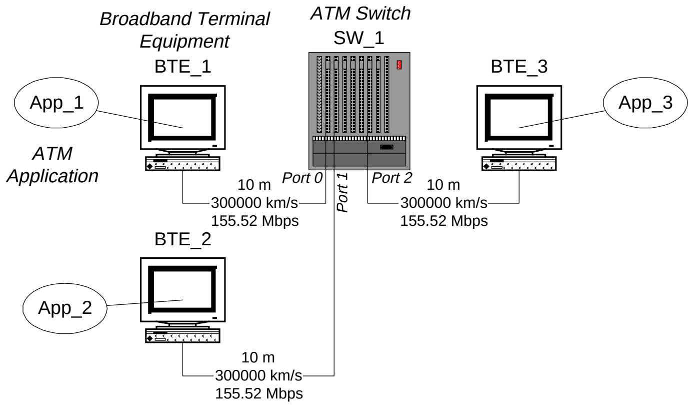

Figure 7: Simulated ATM network.

All the buffers of BTE and switch queueing systems are infinite buffers. The insertion of ATM OAM cells in the physical layer is disabled. The AAL5 packetization delay is 1nseg. The distance between the network elements is 10 meters. The signal propagation velocity in the optical fiber is considered as 300000 Km/sec, which results in a propagation delay of 33.33 nsec. The data transmission rate for all network links is 155.52 Mbps (OC-3) and the switch cycle time is 2.7263 μsec.

5. Simulation Results

In order to demonstrate the simulator results two simulations were performed: one for transient behavior and the other to observe the network steady state. In the transient behavior simulation were performed 1 ms of simulation time and in the steady state the simulator runs 5.5 s of simulation time. The following sections show the simulation results.

5.1.1 Transient Behavior

Figure 8 shows the traffic pattern generated by the applications App_1 and App_2. Figure 9 shows the BTE_1 and BTE_2 queueing system occupancy. Figure 10 shows the SW_1 output buffer occupation for the ATM and physical layers. Figure 11 shows BTE_1 and BTE_2 queueing system waiting time. Figure 12 shows the SW_1 output buffer waiting time for the ATM and physical layers.

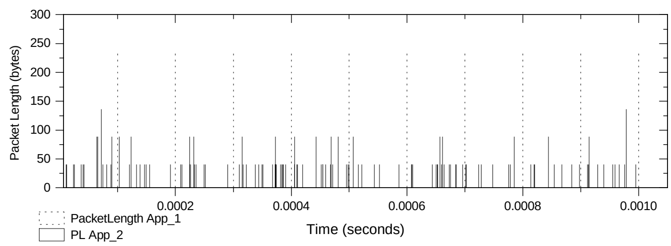

Figure 8: Traffic pattern of applications App_1 and App_2

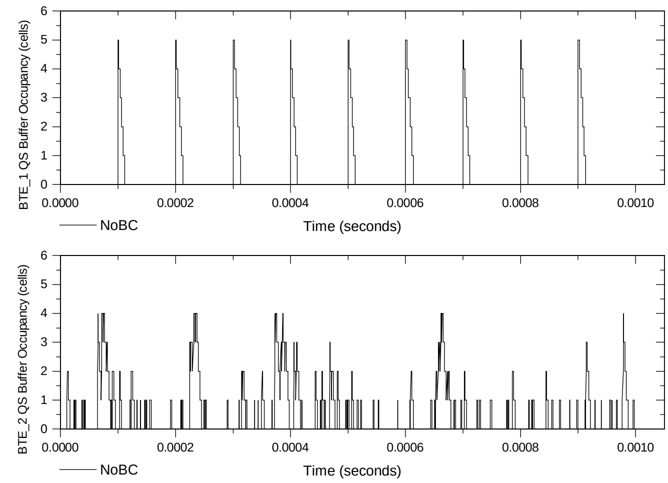

Figure 9: Occupation of BTE_1 and BTE_2 queueing systems buffers.

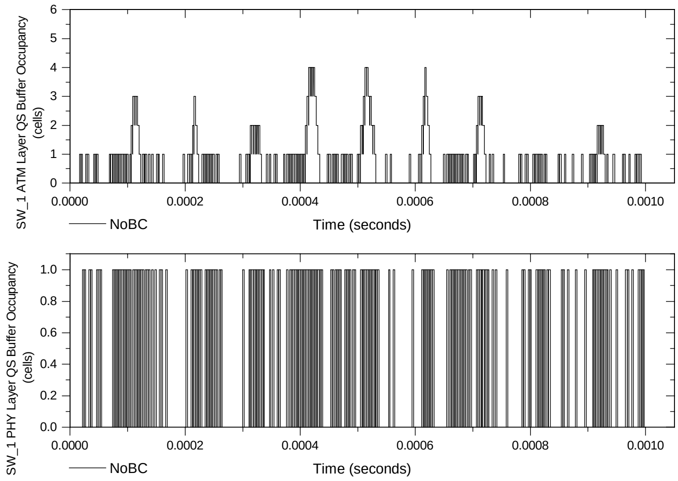

Figure 10: Occupation of SW_1 port 2 queueing system output buffers (ATM and physical layers).

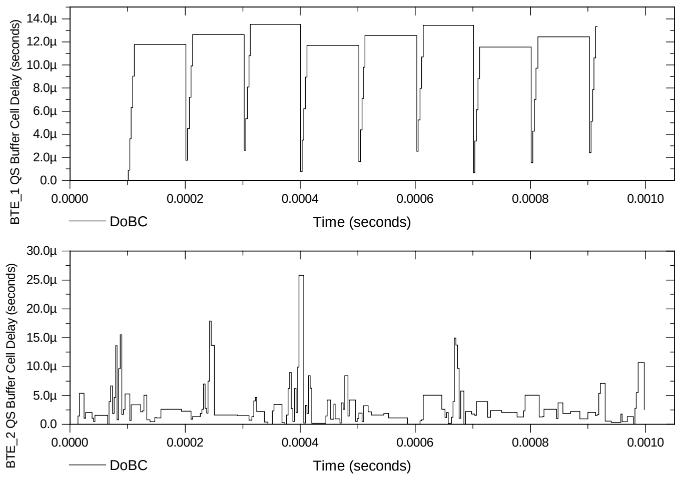

Figure 11: Waiting time in BTE_1 and BTE_2 queueing system buffers.

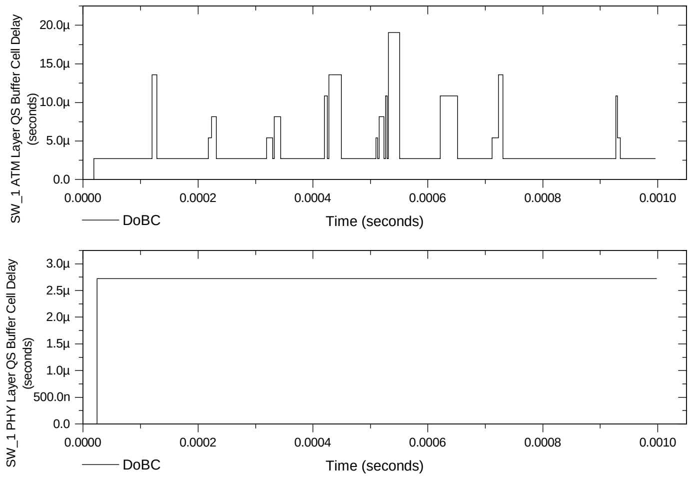

Figure 12: Waiting time in SW_1 port 2 queueing system output buffers (ATM and physical layers).

5.1.2 Permanent Behavior

Figure 13 shows the mean number of cells in the network buffers. Figure 14 shows the waiting time in the network buffers. Figure 15 shows the utilization of network queueing system. Figure 16 depicts simulator performance in steady state.

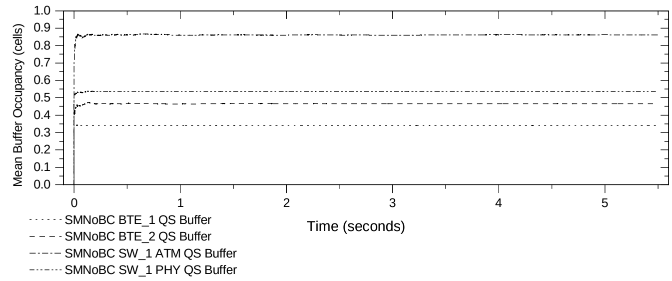

Figure 13: Network buffers mean number of cells.

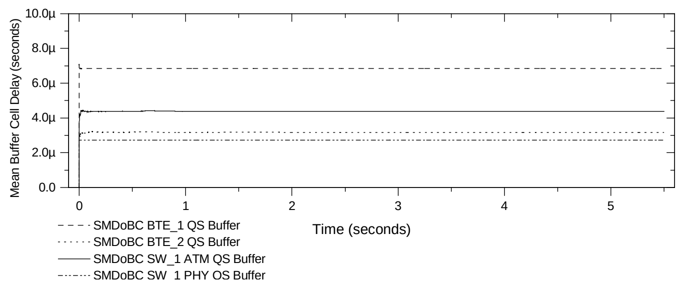

Figure 14: Network buffers mean cell waiting time.

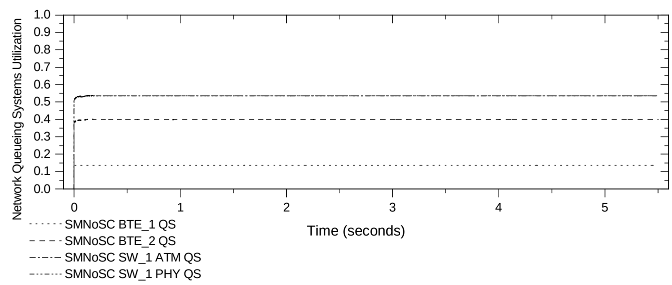

Figure 15: Network queueing systems utilization.

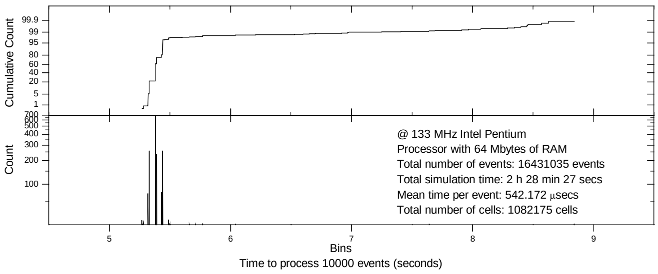

Figure 16: Simulator performance in steady state.

The steady state results showed in the figures can be verified using the queueing system theory [13].

6. Conclusions

The SimATM is a tool developed to ATM networks research, analysis and design, and also for ATM traffic model validation. SimATM is developed in C++ using the benefits of object-oriented simulation, namely modularity, improved reliability and reusability.

The simulator is developed using the event-driven simulation technique with the simulation performed at the ATM cell level. The simulator has extensive simulation statistical data gathering, which allows the simulations analysis even in network transitory or stationary state. The simulator enables the direct comparison of simulation results with queueing system models of the queueing theory.

The models for equipment, applications, layers and queueing systems are fully detailed and very flexible. Future SimATM releases will implement the existing models and other new models using the Windows' Dynamic Link Libraries (DLLs) that permit model substitution without further program recompilation. In future releases new layer models will be added to the simulation kernel. The presented simulation example is a sound demonstration of SimATM simulation resources. The SimATM is supporting researchers in broadband traffic model to develop a realistic model for self-similar ETHERNET LAN traffic inside SimATM.

7. Acknowledgments

We would like to thank you FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) by their support to this project.

8. References

[1] Uyless D. Black, "ATM: Foundation for Broadband Networks", Prentice-Hall, 1995.

[2] Daniel Minolli, Anthony Alles, "LAN, ATM, and LAN Emulation Technologies", Artech House, 1996.

[3] George C. Sackett, Christopher Metz, "ATM and Multiprotocol Networking", McGraw Hill, January 1997.

[4] Anthony Alles, "ATM Internetworking", White Paper, Cisco Systems, Inc., May 1995.

[5] B. Stroustrup, "The C++ Programmig Language", Addison-Wesley, 1995.

[6] J. Klein, L. S. Mendes, "SimNT – User Manual - Version 0.3", Electrical and Computer Engineering School, State University of Campinas, March 1997.

[7] M. R. N. Ribeiro, H. Waldman, J. Klein, L. S. Mendes, "Error Rate Patterns for Modeling of Optically Amplified Transmission Systems", IEEE Journal of Selected Areas in Communications, vol. 15, no. 4, pp. 707-716, May 1997.

[8] ITU-T Recommendation I.321, "B-ISDN Protocol Reference Model and Its Application", Geneva, 1991.

[9] ITU-T Recommendation I.731, "Types and General Characteristics of ATM Equipment", Geneva, March 1996.

[10] ITU-T Recommendation I.732, "Functional Characteristics of ATM Equipment", Geneva, March 1996.

[11] ITU-T Recommendations I.751, "Asynchronous Transfer Mode Management of the Network Element View", Geneva, 1996.

[12] San K. Shanmugan, "Simulation and Implementation Tools for Signal Processing and Communication Systems", IEEE Communications Magazine, July 1994.

[13] Leonard Kleinrock, "Queueing Systems – Volume I: Theory", John Wiley & Sons, 1975.

[14] ITU-T Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) Specification", Helsinky, 1993.

[15] ITU-T Recommendation I.361, "B-ISDN ATM Layer Specification", Helsinki, 1993.

[16] ITU-T Recommendation I.150, "B-ISDN Asynchronous Transfer Mode Functional Characteristics", Helsinky, 1995.

[17] ITU-T Recommendation I.610, "B-ISDN Operation and Maintenance Principles and Functions", Helsinky, 1995.

[18] ITU-T Recommendation I.432, "B-ISDN User-Network Interface – Physical Layer Specification", Helsinky, 1993.