Development of a Modeling Framework to Evaluate the QoS of ATM Connections

A. M. Alberti, F. Sakuray, M. L. Proença Jr., M. Bottoli, D. Zaccarias, S. Penha and L. S. Mendes.

State University of Campinas, Campinas, SP, Brazil.

Abstract – **This paper presents the development of a modeling framework to evaluate the quality of service of ATM switched virtual connections (SVCs). The proposed modeling framework not only encloses all ATM Traffic Management functions and their complex relationship, but also encloses ATM cell transport/processing, ATM traffic contract negotiation, and ATM SVCs routing and management. The modeling framework was implemented in an expansible communication systems simulation environment called SimNT 2.0, which was developed at State University of Campinas. The paper also presents a simulation example that briefly demonstrates how our modeling framework can evaluate the QoS of ATM connections.**

I. Introduction

ATM is being considered the most favorable technology to offer end-to-end quality of service (QoS) guarantees for a wide number of applications such as telephony, desktop video conferencing, video on demand and multimedia based distance learning. Two reasons for that are the native ATM support for QoS and the ATM capacity to cover all portions of the network. To achieve this end-to-end QoS support, ATM relies in a very sophisticated set of Traffic Management (TM) functions, which was defined in both ATM Forum Traffic Management Specification 4.0 [1] and ITU-T I.371 Traffic Contract and Congestion Control in B-ISDN Standard [2]. Although both documents cover a common set of features, each one with some unique functions and terminology, this work is based on the TM 4.0 specification.

Formally, the following set of traffic and congestion functions are specified in TM 4.0: Connection Admission Control (CAC), Usage Parameter Control (UPC), Selective Cell Discarding, Traffic Shaping, Explicit Forward Congestion Control, Resource Management using Virtual Paths, Frame Discard, Generic Flow Control and ABR Flow Control.

These TM functions are designed to prevent network and end-systems from becoming congested. If congestion occurs, TM functions can react in such way to maintain QoS objectives while maximizing the use of network resources. Therefore, with an adequate traffic management it is possible to offer a satisfactory level of QoS to network connections without reducing network efficiency.

In fact, as Giroux et al. [3], the complexity of these TM functions is not intrinsic to ATM; it is required for any technology that aspires to carry traffic efficiently while meeting QoS guarantees. In our vision, the ATM traffic management complexity is aggravated by another important factor: the strong relationship existent among these TM functions. For example, consider the case of offering a certain level of cell loss ratio (CLR) guarantee to an ATM connection. The achieved CLR will depend mainly of the following components:

- The queuing structure adopted to store cells. A queuing structure with individual queues per connection (per-VC) can achieve a better traffic isolation than a first-in first-out queuing [4].
- The queuing structure management used to efficiently shares the available buffer space. Some of the buffer-partitioning schemes offer isolation naturally, while others need to be coupled with a more intelligent discard policy [3].
- The scheduling mechanism adopted to appropriately select the order in which cells should be served. Several scheduling mechanisms can be implemented offering different levels of traffic isolation, delay and throughput bounds and worst-case fairness index (WFI) [5].
- The connection admission control used to decide if a new connection can be accepted. An efficient CAC produces maximum statistical gain without violating QoS guarantees. In general, CAC allocations are used to manage queuing structures and schedulers resources [6].
- The cell discard policy used to discard cells when congestion occurs.
- The traffic policing function used to prevent connections to utilize more resources than negotiated.
- The traffic shaping function used to modify traffic characteristics to conform to contracted traffic descriptors.

Thus, a good estimation of the QoS level in ATM connections must account for all of these TM functions

A. M. Alberti, F. Sakuray, M. L. Bottoli, D. Zaccarias, S. Penha and L. S. Mendes are in the Communications Department of the School of Electrical and Computer Engineering, State University of Campinas, SP, Brazil. M. L. Proença is in the Computer Science Department, State University of Londrina, PR, Brazil. Phone: +55 19 3788 3829. E-mails: {alberti, lmendes}@decom.fee.unicamp.br.

and captures their complex relationship. To evaluate the QoS of ATM SVCs in a flexible, efficient and integrated way, this paper presents a modeling framework that not only encloses all ATM TM functions and their relationship, but also encloses ATM cell transport/processing, ATM traffic contract negotiation, and ATM SVCs routing and management.

The proposed modeling framework was implemented in a communication systems simulation environment called SimNT 2.0. This allows our modeling framework to be easily expanded and, although originally developed for SimNT 2.0, its ideas and solutions can be easily implemented in other simulators. Thus, this paper presents a new generic approach to simulate ATM TM functions in an event-driven simulator.

The remaining of the paper is divided as follows: section II describes SimNT 2.0 simulation environment; section III presents our modeling framework; section IV presents an ATM network simulation example; finally, we draw some conclusions about the modeling framework.

II. SIMULATION ENVIRONMENT

In this section we present SimNT 2.0, which was choosen to implement our modeling framework. SimNT 2.0 was developed in the Communications Department of the School of Electrical and Computer Engineering, State University of Campinas [7], to allow the analysis of multi-protocol communications systems in an integrated way. It is based on event-driven simulation, and was developed in C++ for the Windows TM operating system. SimNT's structure can be divided in two parts: executable program and models library (see Figure 1). The executable program is the main SimNT 2.0 part. It has a simulator's kernel instance. The models library is the set of all models that can be used to simulate a network. In SimNT 2.0 both models and kernel are implemented as Windows TM DLLs – Dynamic Link Libraries.

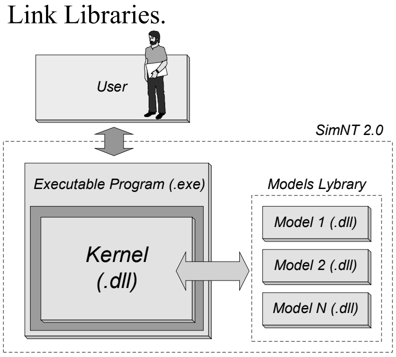

Figure 1. SimNT 2.0 structure.

A. Models Library

SimNT 2.0 allows its users to implement and incorporate new models to the simulation environment without the need to recompile kernel's code. It also allows hierarchical models development, which is implemented through a hierarchical model structure as illustrated in Figure 2. We used these features to implement our modeling framework in SimNT 2.0.

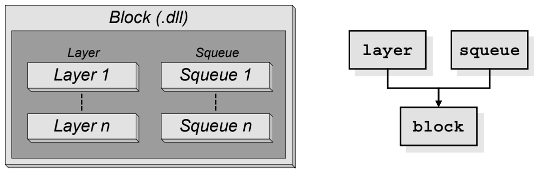

Figure 2. Hierarchical model structure in SimNT 2.0.

This structure is based on three kernel C++ base classes [8]: `Block`, `Layer` and `Squeue`. Classes `Layer` and `Squeue` are the first level of this hierarchical structure, while `Block` class is the second level. Therefore, in SimNT 2.0 the models are built by derivation of `Block`, `Layer` and `Squeue` base classes. Thus, a *block* could have several instances of *layers* and *squeues* (in our modeling framework called *traffic managers*), that are called *block* internal models. It is important to observe that only the *blocks* will be added to the SimNT 2.0 models library as Windows TM DLLs.

B. Connections

Connections are objects used to represent the simulation environment models topological relationship. SimNT 2.0 has a hierarchical connections structure based on three levels:

- First level – *Block connections* (BCs) are used to connect two models (.dll).
- Second level – *Network connections* (NCs) are used to establish a path between two models (.dll). This path is defined as a group of *block connections*.
- Third level – *Data connections* (DCs) are also used to establish a path between two models (.dll), but in this case this path is a logical path defined as a set of *network connections*.

C. Parameters

In SimNT 2.0 parameters are used to configure or to modify model's behavior, allowing the storage not only of scalar values, but also of vectors and matrices.

III. MODELING FRAMEWORK

Our modeling framework (see Figure 3) divides ATM network models in two types: ATM client models and ATM network models.

A. ATM Client Models

These models were developed to allow modeling OSI layers three to seven of a desired ATM client technology. Currently, we have developed two ATM client models: General Application and Receiver Application. The general application is a very flexible model that can be used to manage ATM SVCs, configure ATM traffic contracts and send data through ATM network models. The receiver application was developed to receive packets and to decide when a SVC must be removed at the network's end terminal equipment.

The general application allows loading of real or synthetic traffic patterns, representing ATM client networks aggregated traffic, or external applications traffic, such as NVoD (Near Video over Demand) servers, video cameras or TCP/IP clients, etc. It also allows loading of real or synthetic ATM SVCs requesting and teardown patterns. Figure 3 shows general application structure with its internal *layers*, which can be classified into four categories:

- Connection Requesting and Deleting Layers (CRDLs) – For each ATM Forum service category it is implemented a CRDL. Through parameters selection, users can choose among several layer models for each service category. These layers are capable of requesting and deleting ATM Forum SVCs, which are modeled in SimNT 2.0 as network connections (NCs). Each CRDL provides parameters that allow the configuration of a specific ATM service category traffic contract (see Table 1). CRDLs also provide traffic source layers activation when a DC is established. CRDLs statistics includes DC and NC request blocking probability.
- Connection Ending Layers (CELs) – A single CEL was developed to decide when a SVC must be removed by the appropriate CRDL.
- Traffic Source Layers (TSLs) – Traffic source layers are responsible by the generation of packets traffic in the network. It allows defining for each packet the transmission time and length, which could be deterministic, mathematically calculated (poissonian) or read from file. TSLs statistics include packet time and length for synthetic traffic patterns.
- Traffic Receiver Layers (TRLs) – A single TRL was developed to collect network packets delay statistics.

B. ATM Network Models

These models were developed to model the ATM protocol reference model [9]. Three ATM network models were developed: Broadband Terminal Equipment (BTE), Switch, and Manager (responsibly to route SVCs in the network). These models were developed using cell level modeling approach [10].

The BTE is an ATM edge device model, i.e. a network interface card model. It has the following layer models: ATM layer (BTEATM), ATM adaptation layer (AAL), physical input layer (PHYIN) and output (PHYOUT) layer. Several ATM traffic management functions algorithms and structures [3] are modeled inside BTE: Queuing Structures (QSs); Schedulers (Ss); Connection Admission Control Algorithms (CACs); Buffer Management Algorithms (BMs); Selective Discard Algorithms (SDs); Traffic Police Algorithms (TPs); and Traffic Shaping Algorithms (implemented as scheduler models). As the general application, BTE allows users to choose among several *traffic managers* through parameters selection.

TABLE 1. GENERAL APPLICATION TRAFFIC CONTRACT ELEMENTS.

Traffic Contract Elements	**Service Category**				
	CBR	**rt-VBR**	**nrt-VBR**	**ABR**	**UBR**
Traffic Descriptors	PCR CDVT	PCR CDVT SCR MBS	PCR CDVT SCR MBS	PCR CDVT MCR	PCR CDVT
QoS Parameters	CLR Max-CTD P2P-CDV	CLR Max-CTD P2P-CDV	CLR	-	-
Conformance Definitions	CBR.1	VBR.1, VBR.2 and VBR.3		ABR.1	UBR.1

The switch is a model of an ATM non-blocking switch fabric with a shared memory [3] where virtual output queues are maintained. It has the same *layers* and *traffic managers* of a BTE, with exception of the AAL. It also allows users to choose among several *traffic managers*.

B.1 ATM Adaptation Layer

AAL 5 model does cell segmentation and reassembly. AAL 5 statistics includes network cell loss ratio (CLR); network packet loss ratio (PLR); and network cell transfer delay (CTD).

B.2 ATM Layer

ATM layer functions were divided in two models, one for BTE and other for switch. BTE ATM layer (BTEATM) send cells from AAL 5 to physical layer, and vice-versa. The switch ATM layer (SWATM) switches cells from an input switch port to an output switch port, making use of a switch fabric model (see Figure 3). Switch ATM layer parameters include switch fabric parameters, such as output line rate and cell processing delay. CLR statistics are also collected.

B.3 Physical Layer

The input physical layer sends cells to the ATM layer through the input of the ATM equipment. If the equipment is a switch and the input link rate is higher than the ATM layer switch fabric line rate, queuing is necessary, and the ATM cells are sent first to the associated QS and S models.

The output physical layer sends cells to the next ATM network equipment. If the equipment is a switch and the output link rate is lower than the ATM layer switch fabric line rate, the cells are also sent first to the associated QS and S models. The main parameters for PHYOUT are: output link rate, link length and signal propagation speed. Cell loss ratio statistics are also collected in PHYIN and PHYOUT.

B.4 Queuing Structures (QSs)

In the points where congestion occurs in the network, queuing structures are used to store ATM cells. Two QS models are available:

(1) FIFO (First-In First-Out) – Cells are stored in a single queue, which is serviced by a FCFS scheduler.

(2) Per VC Queuing Structure – Cells are stored in queues per connections, which are serviced by any scheduler model.

Both models have just one parameter: QS capacity (in cells). Statistics include QS occupation and cell delay.

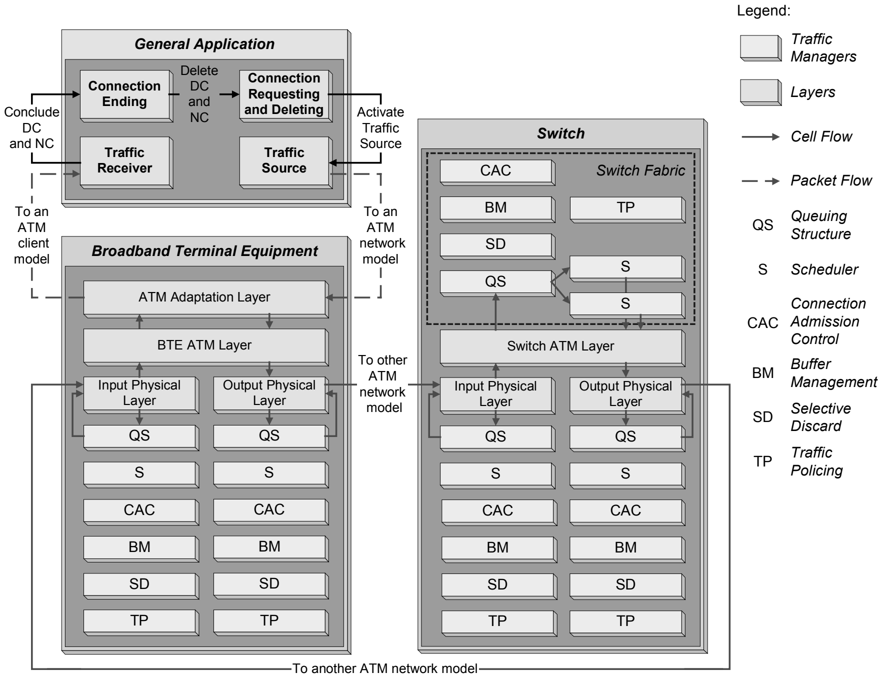

Figure 3. Proposed modeling framework.

B.5 Schedulers (Ss)

Schedulers are implemented in association with each QS to select appropriate service order to ATM cells. We implemented the following scheduling algorithms models:

(1) FCFS (First Coming First Served) – In this model ATM cells are serviced according with its queuing structure arrival time.

(2) PGPS (Packet Generalized Processor Sharing) – This model was developed based on the work of Parekh et al [11]. Cells are serviced according with a virtual service finishing time F_i^k, where k is the k^{th} connection i^{th} cell. For each connection it is allocated a weight ϕ_i.

(3) WF^{2}Q (Worst-case Fair Weighted Fair Queuing) – This model was developed based on the work of Bennett et al [5]. The WF2Q uses a Smallest Eligible virtual Finish time First (SEFF) discipline.

(4) LFVC (Leap Forward Virtual Clock) – This model was developed based on the work of Suri et al [12].

(5) Virtual Scheduling Traffic Shaping Scheduler –This model was developed based on the implementation of virtual scheduling spacers [3]. This model delays received cells until they are conforming to the negotiated traffic contract elements.

The parameter common to all models is scheduler capacity. Statistics include scheduler utilization and bandwidth allocated by CACs models to each ATM SVCs.

B.6 CAC Algorithms (CACs)

At this time, two CAC algorithms are available:

(1) Peak Rate Allocation – This model accepts a new connection if the bandwidth available in the scheduler is sufficient to serve this connection. If a PGPS or a WF2Q scheduler is being used in the same equipment, it adjusts connections weight according with the expression $\phi_i = E/C$, where E is the rate allocated to the connection and C is the scheduler capacity.

(2) Effective Bandwidth Allocation – This model is based on the work of Elwalid et.al [6]. It accepts a new connection if the calculated effective bandwidth and effective buffer are smaller than the available bandwidth in S and the available buffer in QS, respectively. It adjusts connections weight according with the expression $\phi_i = EB/C$, where EB is effective bandwidth allocated and C is the scheduler capacity.

B.7 Buffer Management Algorithms (BMs)

Buffer management algorithms are implemented in each QS to judge if an ATM cell can be stored under a congestion situation. At this time, two BM algorithms were modeled:

(1) Complete Sharing – This model accepts a new cell in the QS only if the total number of scheduled cells plus this cell is smallest than or equal to the QS capacity.

(2) Dynamic Partitioning – This model is based on the work of Krishnan et.al [4], and was developed to work together with the Effective Bandwidth Allocation CAC. It maintains several thresholds (one per VC queuing) that are used to determine if a cell can be accepted or not in a QS. These thresholds are

calculated using the effective bandwidth/buffer allocations calculated in the CAC above.

B.8 Selective Discard Algorithms (SDs)

Selective discard algorithms are used to select and discard lower priority cells in benefit of higher priority cells during congestion situations. We have developed the following SDs algorithms:

(1) Simple Discard – It discards received cells in benefit of QS stored cells.

(2) Cell Loss Priority (CLP) Selective Discard – It discards CLP = 1 cells in benefit of CLP = 0 cells.

(3) Cell Loss Ratio (CLR) Selective Discard – It discards higher CLR connections cells in benefit of lower CLR connections cells.

(4) Service Category Selective Discard – It discards cells according with its service category.

B.9 Traffic Policy Algorithms (TPs)

We developed a leaky bucket traffic police model that is based on the implementation described in [3].

IV. SIMULATION

This section presents a simulation that briefly demonstrates how our modeling framework can be used to evaluate the QoS of ATM connections. Figure 4 shows the simulated network topology. It's composed by: five general source applications (App_0 up to App_4), two BTEs (BTE_0 and BTE_1), one switch (Switch_0), a manager (Manager_0) and a receiver application (App_5).

The example considers a network in two situations:

1. Congested – The CAC algorithm accepts four SVCs that are created by applications: App_0, App_1, App_2 and App_3 (connection 0 up to 3).
2. Severely Congested – The CAC algorithm accepts one extra SVC that is created by application App_4 (connection 4).

Each application transmits a MPEG-4 Simple Program Transport Stream adapted (see [13]) from the traces available in [14][15]. The following *traffic managers* were used in the simulation (see enumerated items from section 3.2.4 up to section 3.2.8): QS = (2), S = (4), CAC = (2), BM = (2) and SD = (3). All source applications establish SVCs to the App_5 using nrt-VBR service category. The ATM traffic contract elements for each application were set according to Table 2, which also shows the weight ϕ_i allocated for each connection by the CAC algorithm. We executed 8 simulations for each situation above. In each of them, the BTE_0, BTE_1 and Switch_0 QSs capacity were set to 16000, 8000, 4000, 2000, 1000, 500, 100 and 50 cells, respectively.

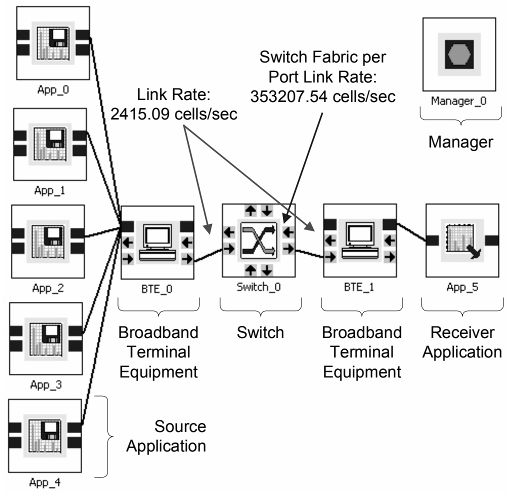

Figure 4. Simulated network topology.

TABLE 2. TRAFFIC CONTRACT ELEMENTS AND SCHEDULER WEIGHTS.

	Trace	PCR (C/S)	CDVT (C/S)	SCR (C/S)	MBS (C)	CLR	ϕ_i
3	Silence of the Lambs	1400	0.0007143	1120	437	1×10^{-6}	0.4637
2	South Park	1191.61	0.0008392	954	52	2×10^{-6}	0.3947
1	Simpson's	1125	0.0008889	900	447	3×10^{-6}	0.3726
4	Die Hard III	1125	0.0008889	900	46	4×10^{-6}	0.3726
0	Futurama	1106.25	0.0009039	885	216	5×10^{-6}	0.3664

A. Results

Figure 5 shows what happens with the QoS of the connections 0 up to 3 when the connection 4 is accepted independently of the CAC recommendation. The following comments can be done for each connection when comparing the situations transition:

Connection 0 – This connection wasn't severely affected by the presence of the connection 4. An interesting effect was the reduction of the QS occupation for capacities larger than 1000 cells (except for 16000 cells). This occurred because connection 0 has the smaller weight among all connections (see the table in Figure 5). As a consequence, the cell delay follows this pattern. However, the CLR is larger than for the situation 1.

Connection 1 – This connection was more affected by the presence of connection 4 than connection 0, especially for small and large buffers. Not only the cell delay but also the CLR increased in situation 2.

Connection 2 – This connection was severely affected by the presence of connection 4, not only in terms of cell delay, but also in terms of CLR.

Connection 3 – This connection was affected by the presence of connection 4 as much as connection 1.

Observe that for all connections, the CLR negotiated is guaranteed only in situation 1 and for QS capacities larger than 16000 cells.

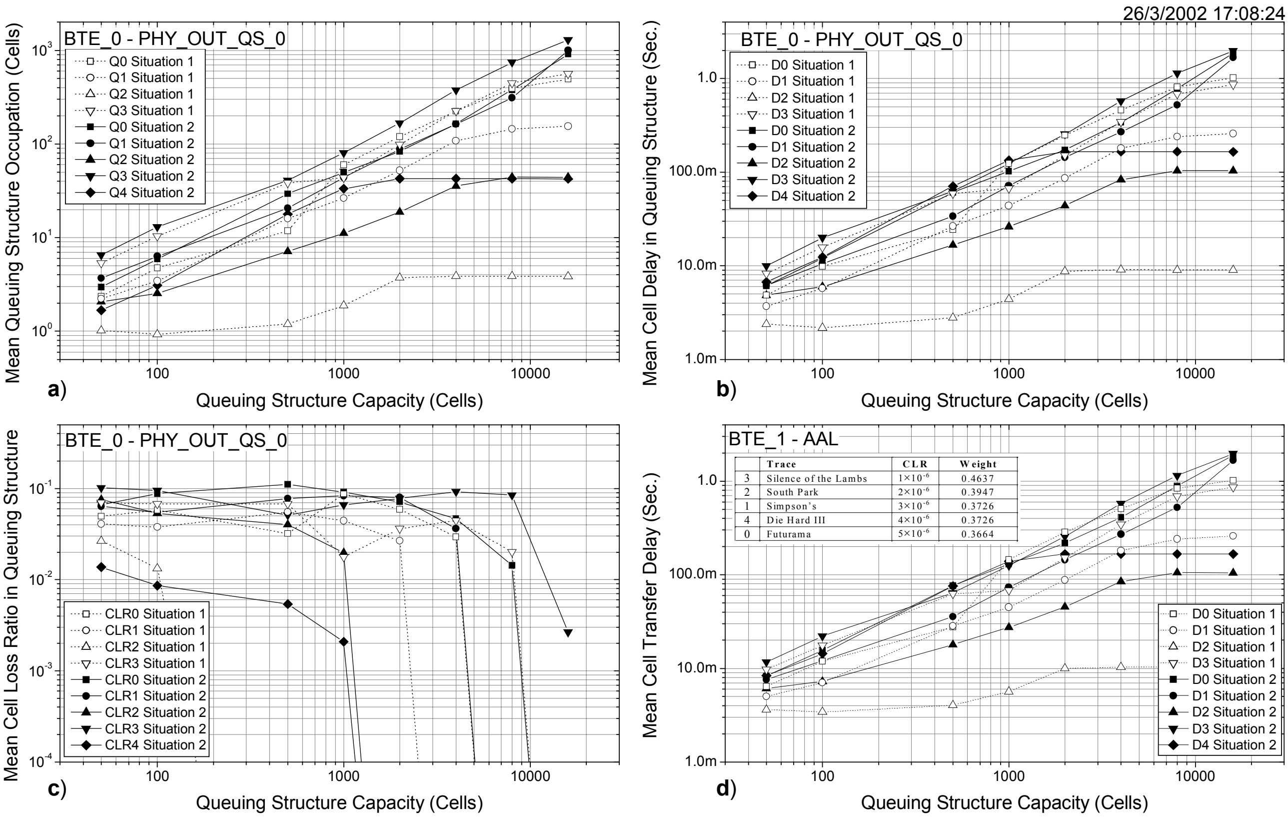

Figure 5. Performance in BTE_0 queuing structure and BTE_1 AAL.

V. CONCLUSIONS

In this paper we presented a modeling framework to evaluate the QoS of ATM SVCs. This modeling framework encloses all ATM Traffic Management functions and their complex relationship, the transport and processing of ATM cells, the negotiation of the ATM traffic contract, and the routing and management of SVCs. The paper also presented a simulation example that briefly demonstrated how our modeling framework could be used to evaluate the QoS of ATM connections under different congestion situations and traffic patterns.

ACKNOWLEDGMENTS

First, we would like to thank the support of FAPESP in the realization of this work. Also we are grateful to TKN Group, Technical University Berlin for making the MPEG-4 video traces available on the Internet.

REFERENCES

[1] ATM Forum, “Traffic Management 4.0”, 1996.

[2] ITU-T, “Traffic Control and Congestion Control in B-ISDN”, 2000.

[3] N. Giroux and S. Ganti, “Quality of Service in ATM Networks: State-of-Art Traffic Management”, Prentice Hall, 1998.

[4] S. Krishnan, A. Choudhury and F. Chiussi, “Dynamic Partitioning: A Mechanism for Shared Memory Management”, IEEE INFOCOM’99, 1999.

[5] J.C.R. Bennett and H. Zhang, “WF2Q: Worst-case Fair Weighted Fair Queuing”, INFOCOM'96, March 1996.

[6] A. Elwalid, D. Mitra and R. Wentworth, “A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node”, IEEE Journal on Selected Areas in Communications, 13(6), 1995.

[7] Ernesto Andrade Neto and Antônio M. Alberti, “Hydragyrum – Ambiente de Simulação de Redes a Eventos Discretos”, SBrT´2000, September 2000.

[8] B. Stroustrup, “The C++ Programming Language”, Third Edition, Addison-Wesley, 1997.

[9] U. Black, “ATM: Foundation for Broadband Networks”, Prentice-Hall, 1995.

[10] G. Kesidis and A. Singh E. C.E, “An Overview of Cell-Level ATM Network Simulation”, High Performance Computing Systems Conference, 1995.

[11] A. Parekh, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case”, IEEE/ACM Transactions on Networking, Vol. 1, No 3, June 1993.

[12] Subhash Suri, George Varghese and Girish Chandranmenon, “Leap Forward Virtual Clock: A New Fair Queuing Scheme with Guaranteed Delays and Throughput Fairness”, IEEE INFOCOM’97, 1997.

[13] A. M. Alberti, E. Andrade and L. Mendes, “Estimativa de Descritores de Tráfego MPEG-4 para o Transporte em Redes ATM Utilizando a Técnica do Buffer Virtual”, XIX Simpósio Brasileiro de Telecomunicações, September 2001.

[14] Frank H. Fitzek and Martin Reisslein, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, IEEE Network Magazine, November 2001.

[15] http://www-tkn.ee.tu-berlin.de/research/trace/trace.html