An Algebraic Model for Evaluating the Performance of an ATM Switch with Explicit Rate Marking

Alessandro Aldini, Marco Bernardo, and Roberto Gorrieri

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
{aldini, bernardo, gorrieri}@cs.unibo.it

Abstract. Stochastically timed process algebras are emerging as a promising formalism to support performance modeling and evaluation of concurrent and distributed systems. To become well accepted, such formalisms have to be proved to be able to provide readable, modular, scalable formal descriptions of more and more complex systems which can be automatically analyzed with a reasonable computational effort. To this aim, we present an algebraic model of an ATM switch which supports the four service categories defined by the ATM Forum and implements explicit rate marking for ABR traffic, backward explicit congestion notification, and connection admission control. As performance measures of interest, we assess via simulation the cell loss ratio, the mean queue length, and the link utilization for the outgoing link in several different configurations.

1 Introduction

Many computing systems consist of a possibly huge number of components that not only work independently but also communicate with each other from time to time. Examples of such systems are communication protocols, operating systems, embedded control systems for automobiles, airplanes and medical equipment, railway signaling systems, air traffic control systems, distributed systems and algorithms, computer architectures, and integrated circuits.

The catastrophic consequences of failures, such as loss of human lives, environmental damages, and financial losses, in many of these critical systems compel computer scientists and engineers to develop formal description techniques for ensuring that these systems are implemented correctly despite of their complexity.

Although a number of description techniques and related software tools have been developed to support the modeling and verification of functional properties of systems, only in recent years formal methods for performance evaluation have received attention. The main benefit of a formal method which allows for the specification of both functional and performance characteristics is that additional project costs due to the late discovery of inefficiency can be avoided.

Moreover, a formal description technique for performance evaluation could provide that linguistic support which is usually lacking. In fact, performance is typically modeled by resorting to stochastic processes such as Markov chains or graphical formalisms such as stochastic Petri nets [1], which yield low level models hard to understand as far as complex systems are concerned, or notations such as queueing networks, which are more abstract but do not allow all the important aspects of a system to be modeled in a formal way. From the designer viewpoint, it would be extremely advantageous to profit from a general purpose description technique resulting in readable performance models, possibly specified in a compositional way, which are unambiguous and can be formally analyzed in an efficient way.

Stochastically timed process algebras (see e.g. [14, 16, 11]) are emerging as a promising formalism to support performance modeling and evaluation of concurrent and distributed systems. Like classical process algebras, they are algebraic languages endowed with a small set of powerful operators whereby it is possible to systematically construct algebraic models from simpler ones. Moreover, stochastically timed process algebras comprise a family of actions which permit to express both the type and the duration of the activities executed by the systems being modeled. As a consequence, the (formally defined) integrated semantic models (such as action labeled transition systems) underlying algebraic terms contain both functional and performance information, so from them projected models (such as action type labeled transition systems and Markov chains) can be (automatically) derived which are guaranteed to be consistent with each other and are useful to investigate functional and performance characteristics, respectively, such as deadlock freeness and resource utilization. Also integrated semantic models are useful from the analysis standpoint as they allow to investigate mixed characteristics, such as mean time to deadlock, which refer to both functionality and performance. Stochastically timed process algebras turn out to be advantageous also because of the notions of equivalence they are equipped with, which relate terms describing systems with the same functional and performance properties and allow for compositional algebraic manipulations.

To become well accepted, stochastically timed process algebras have to be proved to be suitable to describe and analyze more and more complex systems. With respect to this, we believe that it is a useful experiment to see to what extent the performance of ATM networks can be modeled and analyzed by means of stochastically timed process algebras. Actually, such an experiment initiated in [9], where the performance of a simplified ATM switch supporting only ABR (with binary feedback) and UBR traffic was evaluated. The purpose of this paper is to demonstrate that it is possible to model and analyze a more realistic ATM switch which supports the four service categories (CBR, VBR, ABR, UBR) defined by the ATM Forum and implements some of the most important traffic feedback control schemes. The ATM switch is modeled with $\mathrm{EMPA}_{\mathrm{vp}}$ [9], a stochastically timed process algebra with value passing capabilities we have chosen because it yields more compact algebraic descriptions and more compact semantic models. As performance measures, we evaluate the cell loss ratio, the queue length, and the link utilization for the outgoing link in several different configurations. This is carried out by simulative analysis conducted through the EMPA_{vp} based software tool TwoTowers [10].

We have chosen ATM switches for our case study because the ATM technology has been widely studied in recent years and is complex enough to be used as a valid benchmark for evaluating the expressiveness and the adequacy of the stochastically timed process algebraic formalism. As we shall see, our modeling experiment concerning an ATM switch whose characteristics are as described above has emphasized a better readability (w.r.t. other formalisms) of the switch description stemming from the linguistic compositionality of stochastically timed process algebras, the model compactness achieved by enhancing stochastically timed process algebras with value passing capabilities, and the scalability of this description for a wide range of configurations with minor effort.

This paper is organized as follows. In Sect. 2 we briefly recall EMPA_{vp}. In Sect. 3 we give an informal description of the ATM switch we are interested in. In Sect. 4 we model the ATM switch with EMPA_{vp} (see [6] for full details). In Sect. 5 we provide some simulation results obtained using TwoTowers. Finally, in Sect. 6 we draw some conclusions about the adequacy of the stochastically timed process algebra approach.

2 Extended Markovian Process Algebra

2.1 The Language

In this section we recall the operators of EMPA_{vp} we shall use when modeling the ATM switch. For a full presentation of the algebra and its formal semantics, the interested reader is referred to [9].
The constant definition

$$A(type_1\, par_1, \dots, type_n\, par_n; type_{n+1}\, var_1, \dots, type_{n+m}\, var_m) \stackrel{\Delta}{=} E$$

represents the definition of a constant A with certain typed formal parameters and certain typed local variables whose behavior is specified by term E possibly containing invocations of A itself thereby allowing for recursion. As we shall see in the model of the ATM switch, an EMPA_{vp} specification of a given system consists of several constant definitions with one of them designated as the root. Let us now see which form term E can take on.

The null term "$\underline{0}$" is the term that cannot execute any action.

Term $<\alpha, \tilde{\lambda}>.E$ can execute action $<\alpha, \tilde{\lambda}>$ and then behaves as term E. Each action is composed of an action type α and an action rate $\tilde{\lambda}$. Based on their visibility, actions are classified as observable or invisible (denoted by action type τ) depending on whether they can be seen or not by an external observer. According to action types, we have input actions of the form $<a?(\underline{x}), \tilde{\lambda}>$ where $\underline{x}$ is a vector of variables, output actions of the form $<a!(\underline{e}), \tilde{\lambda}>$ where $\underline{e}$ is a vector of expressions, and unstructured actions of the form $<a, \tilde{\lambda}>$ which carry no values. Input/output actions are useful for modeling value passing among terms representing cooperating system components. According to action rates, we have exponentially timed actions of the form $<\alpha, \lambda>$ where $\lambda \in \mathbb{R}_+$ is interpreted as the parameter of an exponential distribution, immediate actions of the form $<\alpha, \infty_{l,w}>$ which take no time and are equipped with a priority level l and a weight w, and passive actions of the form $<\alpha, *>$ whose duration is fixed only upon synchronization with a nonpassive action of the same type.

Term $E[\varphi]$ behaves as term E except that the type of each executed action is modified according to function φ. This operator provides a means to obtain more compact algebraic descriptions.

Term $E_1 + E_2$ behaves as either term E_1 or term E_2 depending on whether an action of E_1 or an action of E_2 is executed first. If the actions executable by E_1 and E_2 are exponentially timed, then the choice is resolved according to the race policy: the action sampling the least duration succeeds. If immediate actions come into play, they take precedence over exponentially timed actions and the choice is resolved according to the preselection policy: the immediate actions with lower priority are discarded, then each of the remaining actions is given an execution probability proportional to its weight. If only passive actions are involved (meaning that the whole term is not completely specified from the performance standpoint), then the choice is purely nondeterministic.

Term *if* (β) *then* E_1 *else* E_2 behaves as either E_1 or E_2 depending on whether boolean expression β is satisfied or not.

Term $E_1 \parallel_S E_2$ asynchronously executes actions of E_1 or E_2 whose type does not belong to S and synchronously executes actions of E_1 and E_2 whose type belongs to S. Two actions can synchronize only if they have the same observable type and at most one of them is nonpassive.

Finally, the constant invocation $A(\underline{e})$, where $\underline{e}$ is a vector of actual parameters, represents the call for the constant definition of A together with the related parameter substitution.

2.2 The Tool

Based on $\mathrm{EMPA_{vp}}$, a software tool called TwoTowers [10] has been implemented. Such a tool automatically translates $\mathrm{EMPA_{vp}}$ specifications into the corresponding semantic models, then these models are automatically analyzed to verify the correctness of systems or to evaluate their performance.

As far as the study of the ATM switch is concerned, the related algebraic model will contain no exponentially timed actions for reasons that will be clarified later. As a consequence, for our case study TwoTowers will generate a discrete time Markov chain as underlying performance model. Since such a Markov chain for an ATM switch is composed of millions of states while our tool only deals with hundreds of thousands of states, we shall resort to the simulation analysis routine of TwoTowers. Such a routine is based on the method of independent replications [26], which consists of statistically inferring performance measures from several independent executions of the specification under consideration. At each step of each execution, only the current state of the specification is needed, the set of its possible immediate transitions is computed, and one of them (together with the related derivative state) is chosen according to the preselection policy as follows. After partitioning $\mathbb{R}_{[0,1]}$ in as many subintervals as there are enabled immediate transitions where the length of any subinterval is proportional to the weight of the corresponding enabled immediate transition, a pseudo random number generator [22] is used to sample a number uniformly distributed in $\mathbb{R}_{[0,1]}$ which determines the immediate transition to execute depending on the subinterval to which it belongs. Since at any time during the simulation only the current state is needed, we are not required to build the whole state space a priori.

3 ATM Switch with Explicit Rate Marking

Asynchronous Transfer Mode (ATM) [23] has been accepted universally as the transfer mode of choice for Broadband Integrated Services Digital Networks (B-ISDN). ATM is a cell switching, asynchronous time division multiplexing, connection oriented technique that requires information to be buffered and then placed in a packet called cell having a fixed length payload (48 bytes) and a fixed length header (5 bytes). ATM delivers important advantages over existing technologies, including the promise of scalable bandwidths and Quality of Service (QoS) guarantees, which facilitate new classes of applications such as multimedia.

An ATM network consists of a set of ATM switches interconnected by point-to-point ATM links or interfaces. Since ATM networks are connection oriented, which means that a virtual circuit needs to be set up across the network prior to any data transfer, the switching elements have predefined routing tables to minimize the complexity of single switch routing. An ATM switching node transports cells from its incoming links (inlets) to its outgoing links (outlets) via the Cell Switch Fabric (CSF) using the routing information contained in the cell header and information stored at each switching node during the connection setup. ATM switches are generally synchronous in the sense that, during a cycle, one cell is taken from any inlet (if one is present), passed into the internal switching fabric, and eventually transmitted on the appropriate outlet. Cells actually arrive on the inlets asynchronously, so there is a master clock that marks the beginning of a cycle. Any cell fully arrived when the clock ticks is eligible for switching during that cycle. A cell not fully arrived has to wait until the next cycle.

ATM can support applications with distinct requirements for bandwidth, throughput, and transmission delay. To fulfil this set of needs, the ATM Forum has defined a family of service categories [7], explained below:

- The Constant Bit Rate (CBR) is the highest priority category, designed for traffic that must meet strict throughput and delay requirements. CBR service is specifically for real time voice and video traffic. CBR connections must guarantee throughput with minimal cell loss and low variation in cell delay. Data is transmitted over the link at the requested rate – no more and, in most cases, no less.
- The Variable Bit Rate (VBR) is used to send at a rate varying from the Minimum Cell Rate (MCR) to the Peak Cell Rate (PCR), and designed for

applications whose information transfer is bursty. As with CBR, VBR users receive guaranteed QoS.
- The Available Bit Rate (ABR) lets ATM users access idle bandwidth to transmit delay tolerant traffic. ABR exploits excess network bandwidth, but it uses traffic management techniques to gauge network congestion and avoid cell loss. ABR is the service in which the network provides rate feedback to the sender, asking it to slow down when congestion occurs. Assuming that the sender complies with such requests, cell loss for ABR traffic is expected to be low.
- The Unspecified Bit Rate (UBR) was intended for applications, like file transfers, with minimal service requirements. UBR provides no specified bit rate, no traffic parameters, and no QoS guarantees. Such connections are not rejected on the basis of bandwidth shortage. Even if cells are lost, the sources are not expected to reduce their cell rate. UBR peculiar features are its lack of flow control and inability to take other traffic types into account.

The ABR congestion management system is made possible by the ATM Forum rate based flow control specification [7]. Feedback from network switches to the end systems gives users the information necessary to respond, by appropriately modifying their submission rates, to changes in the available bandwidth, so that congestion is controlled or avoided. With the Explicit Rate Marking (ERM) [7], the feedback is sent to the sources via Resource Management (RM) cells, and fundamentally consists of three components: the Congestion Indication (CI) bit (if set to 1, speed rate has to decrease), the No Increase (NI) bit (if set to 1, speed rate cannot increase), and the Explicit Rate (ER) field containing rate adjustment information. The ER feedback scheme we consider is a version of the ER Indication for Congestion Avoidance (ERICA) [19].

ER switches normally wait for the arrival of an RM cell to give feedback to a source. However, under extreme congestion, they are allowed to generate an RM cell and send it immediately back to the source. This optional mechanism is called Backward Explicit Congestion Notification (BECN).

In order for a broadband network based on ATM to achieve a high level of performance, more traffic control capabilities have to be introduced. The Connection Admission Control (CAC) is a set of actions, taken by the network at the call setup phase, that determine whether to accept or reject (non UBR) ATM connections. CAC schemes generally accept calls only when there is enough bandwidth to establish the connection at the requested QoS, without compromising pre-existent connections.

3.1 Related Work

A few formal models of ATM networks can be found in the literature. As an example, in [24] a fairisle ATM switch fabric has been formally modeled using multiway decision graphs in order to verify the correctness of the switch fabric. In [25] the ATM signaling protocol specified by ITU-T in Q.2931 has been formally described in SDL in order to identify and validate different error classes.

In [8] the queueing delay experienced in the output buffer of an ATM switch has been formally analyzed via stochastic Petri nets; the formal model, analytically solved using a decomposition approach, assumes periodic burst arrivals for VBR and Poisson arrival processes for ABR traffic types at the burst level; the primary objective is the study of the trade off between buffer loss probabilities and delay deadline violation probabilities. The traffic model, which is at the burst level rather than at the cell level, allows to obtain fast and efficient numerical solutions. In [21] the knockout switch is modeled using stochastic activity networks (SANs) in order to analyze its performance in terms of cell loss probabilities and distribution of queue length for a wide range of burst parameters; the same system is modeled in [3–5] via Generalized Stochastic Petri Nets (GSPNs) and Stochastic Well-formed Nets (SWNs). In [15] a switch based on a three stage delta network is modeled and simulated via the stochastic process algebra SPADES. Finally, in [2, 3] models of an ATM switch, where users exploit either the UBR or the Relative Rate Marking ABR ATM service category, have been formally described via GSPNs and their performance has been assessed via simulation.

To the best of our knowledge, no prior work has formally analyzed the performance of the congestion control mechanisms for ERM ABR class at the ATM level. Additionally, we would like to point out that we have been able to model the features of the considered ATM switch thanks to the value passing capabilities of $\mathrm{EMPA}_{\mathrm{vp}}$. In particular the value passing capabilities have turned out to be necessary to model the congestion control mechanisms for ERM ABR class as its behavior depends on the value of some parameters.

4 Algebraic Model of an ATM Switch with ERM

The aim of this section is to formally model with $\mathrm{EMPA}_{\mathrm{vp}}$ an ATM switch supporting CBR, VBR, ABR, and UBR traffic and implementing ERM, BECN, and a simplified CAC mechanism, in order to compute the cell loss ratio, the queue length, and the link utilization for a single outlet in several different configurations, so as to evaluate the performance of the version of ERICA we have considered. It is worth noting that the presence of CBR and VBR traffic, as well as of the CAC algorithm, is necessary for a meaningful evaluation of the feedback scheme for the ABR class, which is the primary objective of this study. Accounting for UBR traffic is also useful to emphasize the advantages of supporting rate based flow control mechanisms by presenting some results about configurations where ABR traffic substitutes for UBR traffic. Moreover, we shall make in the sequel some simple assumptions about non ABR traffic and CAC mechanism, which are more than enough if we want to simulate a realistic variable traffic configuration. In particular, we can study the ABR flow control specification in the presence of variable unused bandwidth. Furthermore we are sure congestion depends on ABR and UBR classes only, because other traffic control mechanisms for CBR and VBR classes are operating, so we can directly assess the efficiency and performance of the ERICA algorithm. It is worth noting that we are interested in evaluating the effects on the ABR class of the presence of other classes with higher priority, but not the efficiency and performance of these classes and the related control mechanisms.

The algebraic model will be developed in a top down fashion thanks to the compositional linguistic support provided by $\mathrm{EMPA}_{\mathrm{vp}}$. We make the following assumptions about the ATM switch:

- There are n inlets and n outlets with $n \in \mathbb{N}_+$. The $\mathrm{EMPA}_{\mathrm{vp}}$ model will be developed in such a way that it scales w.r.t. n: it will be possible to add (remove) an arbitrary number of inlets (outlets) with minor changes in the model. We shall analyze n inlets contending for a single outlet to exacerbate link congestion.
- Every inlet supports at most one connection at a time. It is worth noting that we shall model the switch structure, but not the source and destination end systems (SES and DES), because we are not interested in end-to-end metrics. In particular, we shall model the incoming traffic flow but not the source of each connection. As this incoming flow has to change according to the related class of service, we shall model part of the source behavior (such as the rate variations for VBR connections, which depend on source behavior only, and the rate variations for ABR connections, which depend on feedback mechanisms) directly within the inlets.
- The incoming traffic per connection follows a Bernoulli distribution with parameter $p \in \mathbb{R}_{]0,1[}$, i.e. during a clock cycle either a cell arrives with probability p or no cells arrive with probability $1 - p$.
- Every connection is unicast, i.e. involves just one outlet. It would be possible modeling multicasting, but we have chosen to consider unicast connections only for the sake of simplicity.
- The cell switch fabric is nonblocking. Several incoming cells headed to the outlets can be delivered during the same clock cycle without contention.
- Output queueing is implemented, i.e. a buffer for exceeding cells is located in each outlet.
- Every outlet supports at most one CBR (VBR) connection at a time, for the sake of simplicity.
- The switch acts as a virtual destination, i.e. it turns around immediately the RM cells to the sources, so rate variations for ABR connections depend on the feedback of this switch only. In short, the link load level for the switch depends on the incoming traffic (simulated in the inlet model) and the incoming traffic partially depends on the congestion level and general load information of the switch (through the feedback control mechanism simulated in the outlet model).

The full $\mathrm{EMPA}_{\mathrm{vp}}$ specification of the ATM switch can be found in [6].

4.1 Overall Model of the Switch

Table 1 shows the overall model of an ATM switch supporting CBR, VBR, ABR, and UBR traffic and implementing ERM, BECN, and a simplified CAC mechanism according to the assumptions above. We exploit compositionality in order to concentrate on the various components of the switch.

Because of the slotted nature of the transmission, we resort to a discrete time $\mathrm{EMPA_{vp}}$ description of the switch containing only immediate actions. In order to model the abstraction of the time unit, we introduce the action type *clock* on which all the switch components must periodically synchronize. Term *Clock* endlessly performs action $<clock, \infty_{1,1}>$ on which every switch component must synchronize whenever no other local action logically belonging to the same time unit can occur. To enforce this, every term modeling a switch component will enable action $<clock, *>$ at any time, and will consist of actions whose priority level is greater than 1.

Term *Input* is made out of the parallel composition of n independent inlets. Term *Output* consists of three concurrent components for each outlet: a buffer manager $OutQueue(0, max)$, a traffic manager $CheckTraf(1, 0, 0, max)$, and a connection admission control manager $CAC(0, max)$. Note that the relabeling operator of $\mathrm{EMPA_{vp}}$ has allowed us to specify a single term to represent n different outlets (inlets).

Term *CSF* sends cells and other information from inlets to outlets and the feedback information in the opposite direction. The numbers i and j identifying the sending inlet and the receiving outlet are represented through two parameters of an output action (e.g. *arr_cell*) and sent to the *CSF*, which passes this information to outlet j through an output action (e.g. $s\_arr\_cell_j$ with s meaning *switched*), where i has been passed as a parameter (the same thing is valid for other actions). In this way, the inlet and outlet models no longer depend on the number n of implemented ports. If we change n, we only have to modify term *CSF*, because it has to address explicitly a different number of ports, and the overall model of the switch, by modifying accordingly the number of parallel components of *Input* and *Output*.

4.2 Inlet Model

Term *Inlet* decides which kind of class of service will occupy the connection. It is worth noting that we can change the probability distribution for the four classes (e.g., we may be interested in evaluating performance when no CBR and VBR connections are active, or when a VBR connection is active on port i and ABR connections are active on each other port). Now let us focus our attention on the sequence of the events, based on the priority of each immediate action. During a time slot, the following events can occur, in the specified order:

1. Closing of pre-existing connections (*close_conn*, *rate_var*).
2. Opening of new connections (*rate_var*). For ABR class, the transmission of an RM cell will follow; differently, the transmission will start in the next slot.
3. The traffic manager informs the buffer manager about the traffic conditions on the outgoing link (*check_load*).

$$
\begin{aligned}
SwitchER_{NxN} &\triangleq (Input(max) \,\|_{I \cup T}\, (CSF \,\|_S\, Output(max))) \,\|_T\, Clock \\
Input(\text{int } max;) &\triangleq Inlet(1, max)[\varphi_1] \,\|_T \ldots \|_T\, Inlet(n, max)[\varphi_n] \\
Output(\text{int } max;) &\triangleq (OutQueue(0, max) \,\|_J\, CheckTraf(1,0,0,max) \,\|_{K_1}\, CAC(0))\,[\varphi'_1] \\
&\quad \|_T \ldots \|_T \\
&\quad (OutQueue(0, max) \,\|_J\, CheckTraf(1,0,0,max) \,\|_{K_n}\, CAC(0))\,[\varphi'_n]
\end{aligned}
$$

$$
Clock \triangleq <clock, \infty_{1,1}>.Clock \qquad T = \{clock\}
$$

$$
\begin{aligned}
I = \{&arr\_cell,\ arr\_cell\_rm,\ s\_var\_spd\_abr_i,\ red\_becn_i, \\
&rate\_var,\ close\_conn,\ up\_conn_j \mid 1 \le i, j \le n\}
\end{aligned}
$$

$$
J = \{check\_load,\ abr\_var\} \qquad K_j = \{s\_close\_conn_j\}, \quad 1 \le j \le n
$$

$$
S = \{s\_arr\_cell_j,\ s\_arr\_cell\_rm_j,\ s\_close\_conn_j,\ s\_rate\_var_j,\ var\_spd\_abr \mid 1 \le j \le n\}
$$

$$
\varphi_i = \{(s\_var\_spd\_abr,\ s\_var\_spd\_abr_i),\ (red\_becn,\ red\_becn_i)\}, \quad 1 \le i \le n
$$

$$
\begin{aligned}
\varphi'_j = \{&(up\_conn,\ up\_conn_j),\ (s\_close\_conn,\ s\_close\_conn_j),\ (s\_arr\_cell,\ s\_arr\_cell_j), \\
&(s\_arr\_cell\_rm,\ s\_arr\_cell\_rm_j),\ (s\_rate\_var,\ s\_rate\_var_j)\}, \quad 1 \le j \le n
\end{aligned}
$$

Table 1. Overall Model of the Switch

4. The cell transmission starts from inlets to outlets with a cell for each connection, in mutual exclusion for cells contending for the same outlet (*arr_cell*, *arr_cell_rm*). VBR connections decide possible speed changes, e.g. the end of a bursty period (*rate_var*).
5. Buffer management and feedback mechanism control (*abr_var*).
6. ABR connections receive feedback messages and modify their submission rate if necessary (*var_spd_abr*, *red_becn*).

Now let us focus our attention on the different components of *Inlet*, which are related to the four classes of service.

As far as the ABR class is concerned, the congestion control is based on the idea that each sender has a current rate that falls between 0 and *max* (PCR). When congestion occurs, the rate is reduced. When congestion is absent, the rate can be increased. The decrease is determined by the rate decrease factor (RDF) parameter; unlike the increase, which is additive and determined by the rate increase factor (RIF), the decrease is multiplicative (it has been shown that additive increase and multiplicative decrease is sufficient to achieve fairness [13]). The connection request is always accepted (ABR is not submitted to CAC algorithm) since we assume that minimum rate for ABR class is zero, so this type of connection cannot affect the QoS of other pre-existent services. The connection always starts by sending an RM cell; after a transmitted cell, the inlet has to wait for different possible feedback messages, according to the feedback type (normal RM cell or BECN algorithm). Whenever an inlet is enabled to transmission (according to speed rate), it can receive either a request for closing the connection with probability q or a cell with probability $1 - q$. If a cell is enabled to arrive in a given time unit, it does arrive with probability p otherwise *Inlet* is idle, hence we are assuming a Bernoulli distribution for the incoming traffic (where p represents the load factor for the current connection). Note that every Nrm cell should be an RM cell, furthermore we use a parameter Trm which is an upper bound on inter-RM time. If congestion occurs and the speed is already the minimum rate, then the transmission can be stopped for an indeterminate period. If connection is stopped, the source periodically sends an RM cell with probability β; the smaller β is, the greater the gap between two RM cells is. If the connection is active, the source has three possible events: sending a normal cell, sending an RM cell, or staying idle. The decision concerning the cell type depends on the rate r, the load factor p, the ratio between normal and RM cells, and the maximum gap of time between two RM cells. Terms for the other classes are similar to this one.

CBR (VBR) connections are submitted to the CAC algorithm, so they can be refused. It is worth noting that the priority of the refuse event is less than the priority of the accept event, because a connection, if possible, must be accepted. In case of acceptance, the connection is opened at the rate agreed upon, which belongs to the range $1 \ldots max$. As far as the CBR class is concerned, speed will be kept constant throughout the connection. The transmission rate for VBR connections can change owing to the features of the VBR service class. After an incoming cell, if the rate is less than PCR then the rate itself can increase, because a bursty period is starting. After an idle slot, if the rate is greater than MCR then the rate itself can decrease, because a bursty period is ending; so, connections alternate bursty periods and idle periods. It is worth noting that no leaky bucket algorithm is implemented for the VBR class (we assume that the input meets the negotiated parameters). Closing CBR (VBR) connections is modeled by means of a synchronization between the involved inlet and outlet, for two reasons: the traffic manager has to know the bandwidth variation and the CAC algorithm has to be able to accept new requests (CBR or VBR).

The UBR class is not submitted to CAC algorithm and the traffic manager does not consider UBR traffic. All UBR cells are accepted and if there is capacity left over, they will also be delivered. If congestion occurs, UBR cells will be discarded, with no feedback to the sender and no expectation that the sender slows down. The UBR service does not guarantee the QoS as every portion of unused bandwidth is freely exploited without observing any flow control mechanism.

4.3 Outlet Model

As we have seen in Table 1, the outlet model has three components: a CAC manager, a traffic manager, and a buffer manager. The simple way we modeled CAC algorithm allows to implement one of the most complex functions of the cell switch fabric. We can use a simplified model because we are not interested in evaluating CAC performance. The traffic manager has to save each bandwidth variation, then it has to inform the buffer manager about the link congestion level, the type of active traffic on the link, and every other information necessary for the ER indication algorithm. The term $CheckTraf_j$ has three parameters:

- Available rate, i.e. the unused bandwidth available for ABR class:
  $$ABR\ Target\ Capacity = Target\ Utilization \cdot Link\ Rate - VBR - CBR$$
  We assume the link capacity equal to 1 and the target utilization equal to 100%, but this variable can be easily changed by handling the first parameter of *CheckTraf*.
- Input rate, i.e. the portion of bandwidth occupied by the ABR class. The load factor z for the outgoing link is calculated as the ratio of the measured input rate at the port to the target capacity of the output link.
- Number of active ABR connections, used to calculate the fair share of the available link capacity per active ABR virtual circuit flow.

Each action affecting bandwidth variations helps to modify the value of the above parameters. Now let us specify the relation between the rate values (from 1 to max) and the above parameters. For the sake of simplicity we assumed the link capacity equal to 1; this value also represents the weight of the PCR. So a middle rate (represented by the value $max - 1$) has weight 1/2 and so on until the MCR (its value is 1), with weight $1/max$. The function describing this relation is Weight $(r) = 1/(max - r + 1)$. It is worth noting that zero rate has weight $1/(max + 1)$, in fact a stopped connection occupies the link sending RM cells periodically.

The buffer manager has a formal parameter: the queue length. The traffic manager communicates the information about load to the buffer manager every time unit. At each time unit the buffer manager can do nothing if it is empty and there is no incoming cell, send the first cell in the buffer if it is not empty and there is no incoming cell, or accept an incoming cell and send the first cell in the buffer if it is not empty and there is an incoming cell. Since an outlet can be simultaneously connected to several inlets, it can receive several incoming cells during the same time unit, and those cells which cannot be accommodated in the buffer are lost. In this paper we are not interested in the choice of a particular outgoing cell; we are assuming that the outgoing link can accept the first cell in the buffer at each time unit. If ABR connections are active, we manage the buffer according to the queue length; whenever the first threshold is not exceeded the condition $NI = CI = 0$ (no pending congestion) is verified, else if the second threshold is not exceeded pending congestion is verified ($NI = 1$), otherwise congestion occurs ($CI = 1$). For the sake of simplicity this information is used here (with the ER calculated in the ERICA algorithm) for the computation of the new Allowed Cell Rate (ACR) for the current ABR connection (this is possible because no other switch can modify bits NI and CI or reduce the ER calculated). Table 2 summarizes every possible rate control action according to the congestion level (absent, pending, or occurring).

The buffer manager implements the basic algorithm for the Explicit Rate Indication and other innovations, like fairshare first to avoid transient overloads,

CI Bit	NI Bit	Action on Rate
0	0	ACR $\leftarrow$ Min(ER, ACR + additive increase, PCR)
0	1	ACR $\leftarrow$ Min(ER, ACR)
1	–	ACR $\leftarrow$ Min(ER, ACR – multiplicative decrease)

ACR $\leftarrow$ Max(ACR, MCR)

Table 2. ERM. Control feedback

single feedback in a switch interval, ABR operation with VBR and CBR in the background. All boundary cases are treated [19]. The buffer manager reads the ERICA parameters at the beginning of a new time slot. Therefore the switch is allowed to generate BECN cells; these cells may not ask a source to increase its rate (CI bit is set) and are limited to a fixed number per second. BECN algorithm is activated when outlet does not receive RM cells for a fixed time.

5 Performance Analysis Results

In this section we shall report on the use of TwoTowers to evaluate some performance indices of the ERM and ERICA schemes, such as the cell loss ratio, the queue length, and the link utilization for the outgoing outlets for various configurations, source models, and background traffic. We assume that there are n inlets contending for a single outlet; this configuration exacerbates link congestion and allows to verify ERM and ERICA algorithms. The inlets all have the same characteristics ($q = 10^{-6}$), and the outlets all have the same characteristics as well. The three performance indices have been assessed by varying the buffer capacity L, the parameter p of the Bernoulli incoming traffic (we shall refer to this variable as the load factor of the connection), the ratio between each kind of service, the number of implemented rates (default value of max is 10), and other parameters concerning the congestion control mechanism (e.g. the target utilization for ABR service, whose default value is 100%). Performance figures have been obtained through a simulative analysis of the $\mathrm{EMPA_{vp}}$ model of the ATM switch with ERM developed in the previous section. We have simulated the behavior of the switch for one second. Assuming a capacity of 155 Mbps and recalling that every cell is 53 byte long, it turns out that the clock cycle is about 3 μsec, so we have considered $333,000$ clock cycles, i.e. we stopped every simulation run after $333,000$ actions with type *clock* had been executed. For a real configuration of parameters, we assume the list of ABR parameters as in Table 1 in [18] (e.g. Nrm is 32 and Trm is 100 msec), but it is worth noting that if we change opportunely the model parameters, we can reduce the duration of the simulation run, obtaining the same results.

In order to allow a comparative study between the ABR class, supporting feedback control, and the UBR class, we present some interesting results in the following table, concerning the cell loss ratio of some traffic configurations with two inlets contending for the same outlet and no ABR connection.

	Buffer capacity			
CLR	512	1024	4096	16384
VBR + UBR	0.492	0.491	0.488	0.479
CBR + UBR	0.187	0.174	0.167	0.138
UBR + (CBR or VBR)	0.393	0.386	0.38	0.366

We add that the link utilization is always greater than 99.99%. These results can be meaningful for an estimate of the feedback control mechanism performance; in fact, as we shall see, the effects of the ERM mechanism are evident on the cell loss ratio, whereas no appreciable difference can be observed on link utilization. Now let us analyze the performance of the ERM algorithm: the first test is to use a single ABR source configuration. A scheme that does not work for this simple configuration is not worth further analysis. Then we use a multiple source configuration. For every configuration the cell loss ratio is zero, while the link utilization is the following for a varying load factor p:

p	1 ABR	2 ABR	3 ABR	4 ABR
0.85	87.8	87.8	88.5	89.1
0.90	92.0	92.9	93.0	93.4
0.95	96.3	96.6	96.7	96.9
0.99	99.2	99.5	99.7	99.9

As we can see, ERICA and ERM schemes achieve the required efficiency, since the source rate rises to almost fully utilize the link with no cell loss. These results are completely compatible with those in [20], where a complete presentation of the ERICA algorithm can be found; due to lack of space, we do not reproduce the tables concerning link utilization, so the interested reader is referred to [20], page 186 and the following.

The simulation results reported in Fig. 1 to 5 are plotted together with the related 90% confidence intervals which are depicted as vertical bars. These figures show the performance for multiple ABR sources and a single VBR source contending for the same outlet (Fig. 1 refers to a model with 3 sources, fig. 2 to 4 with 4 sources, and fig. 5 with 6 sources). As we can see, the cell loss probability rapidly converges to zero; ABR uses up all the left-over capacity and the link is not underutilized. The queue length (whose figures are plotted for an interval of 10 milliseconds) depends on the variability of the VBR source, the number of ABR sources and the load factor of each connection. Different results achieved by varying the number of ABR sources converge to the same values if we modify opportunely the buffers capacity. These configurations are presented also in [20, 17] with the same results. Fig. 3, 4 show what happens if we decrease the Target Utilization (T.U.) for the ABR service from 100% to 90%: as expected, the former model outperforms the latter as far as throughput

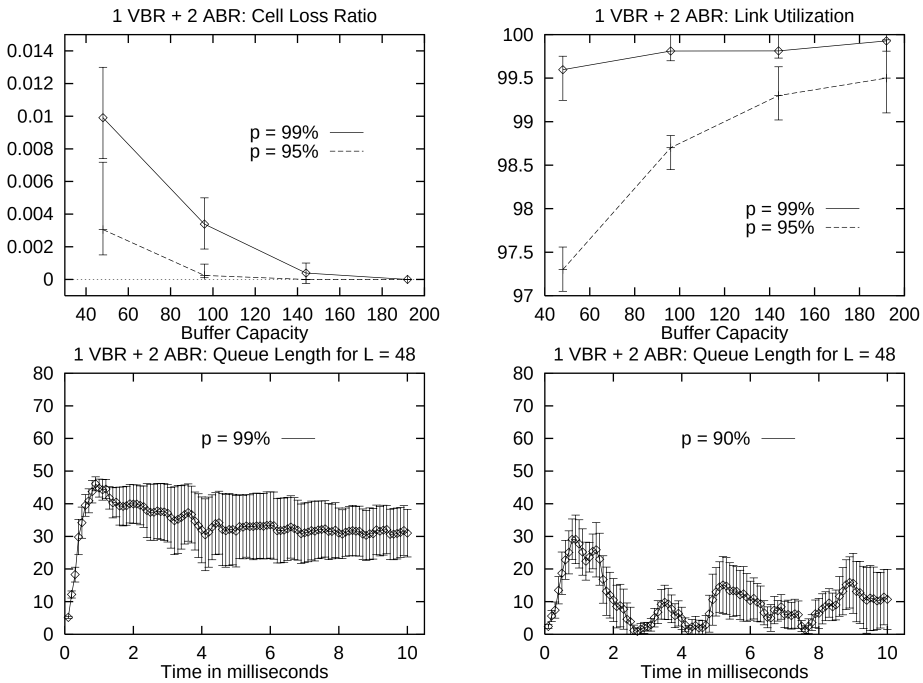

Fig. 1. Results for 1 VBR source and 2 ABR sources.

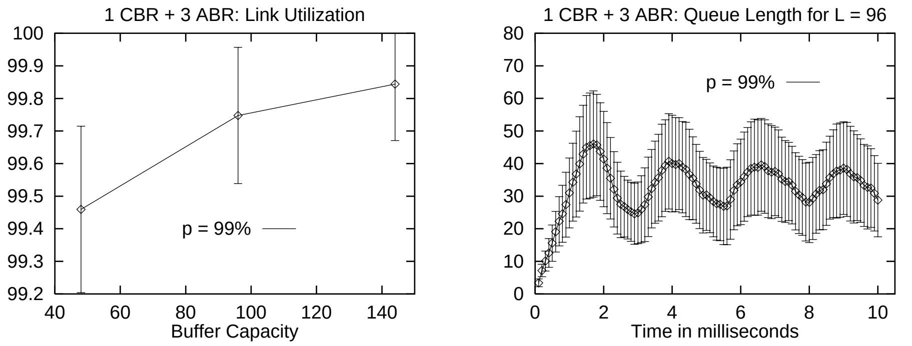

Fig. 2. Results for 1 CBR source and 3 ABR sources.

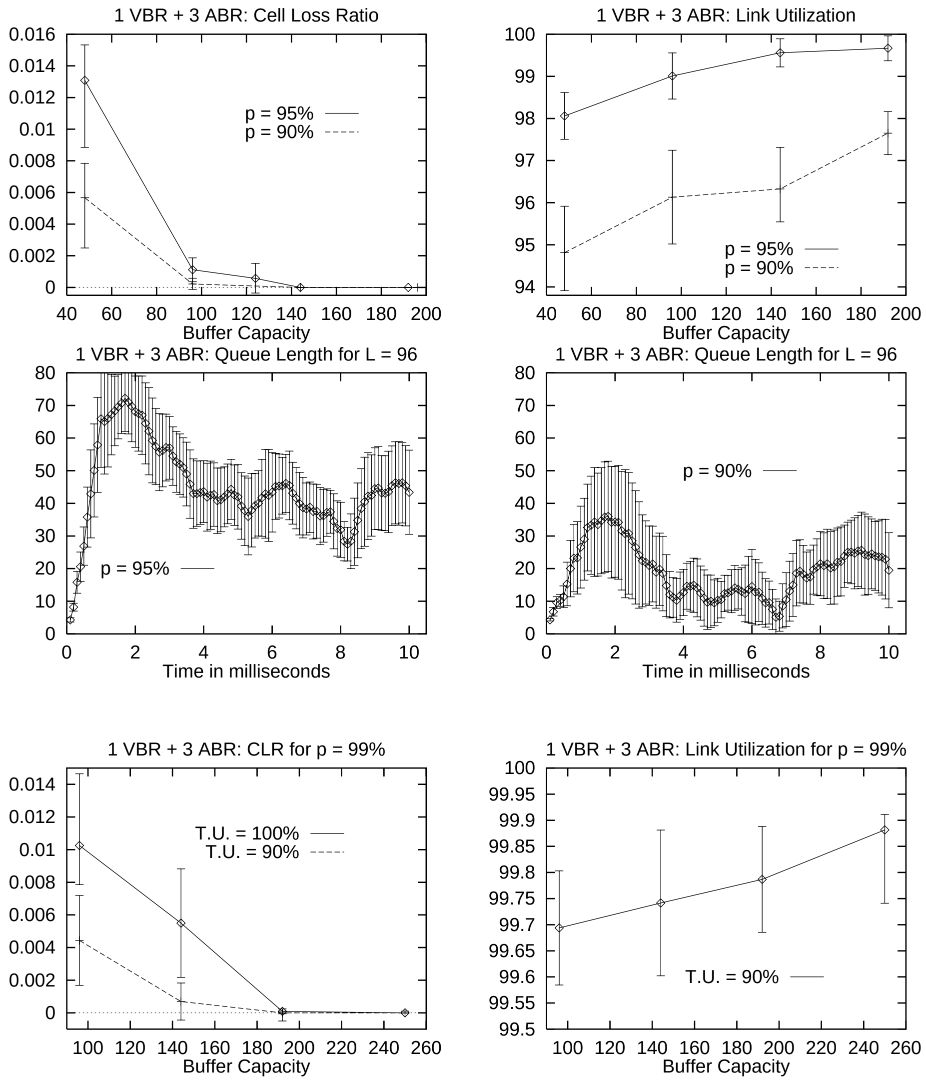

Fig. 3. Results for 1 VBR source and 3 ABR sources.

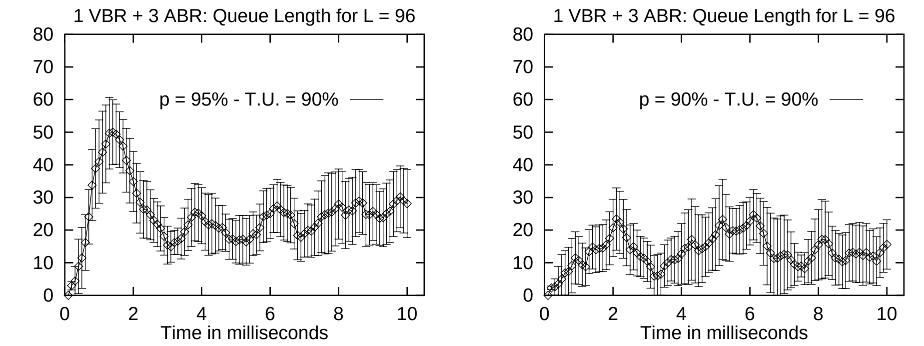

Fig. 4. Results for 1 VBR source and 3 ABR sources (continued).

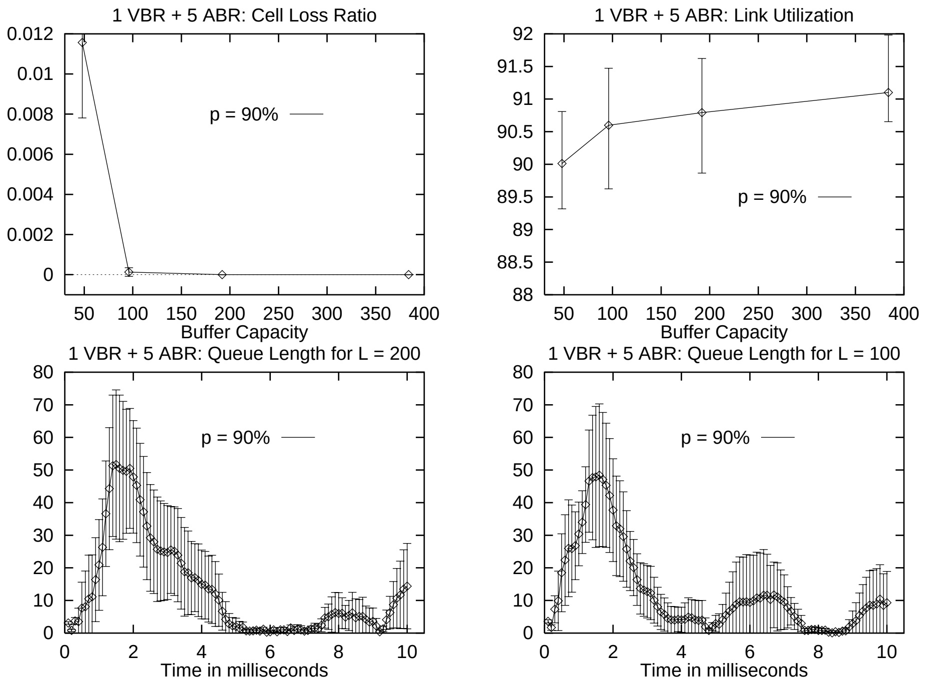

Fig. 5. Results for 1 VBR source and 5 ABR sources.

is concerned (for p = 99% and T.U. = 100%, the link utilization is 100%), but the reverse is true as far as the cell loss ratio is concerned. Varying the model configuration for the above experiments is simple, because the target utilization is a parameter of constant definitions in the $\mathrm{EMPA}_{\mathrm{vp}}$ model of the ATM switch. Fig. 2 shows link utilization and queue length for a configuration with a CBR source instead of a VBR source; it is worth noting that the cell loss ratio is null.

As expected we have noted that if we increase the number of possible outlets, congestion is not exacerbated as in the previous configurations, and the cell loss ratio decrease is appreciable (it is absent without UBR traffic and in any other case it rapidly converges to zero). A comparison between the model of this work and the model described in [9] (ATM switch with EFCI mechanism) confirms that the former algorithm outperforms the latter as far as the cell loss ratio and the link utilization are concerned. Achieved results concern functionality and performance of the feedback traffic control mechanism (ERM and in particular ERICA) and features of ABR service class. Obtained results are relevant, even if in our setting we can evaluate the ATM protocol only, but not the performance of TCP/IP over ATM.

6 Conclusion

In this paper we have formally modeled and analyzed, by means of the stochastically timed process algebra $\mathrm{EMPA}_{\mathrm{vp}}$, an ATM switch which supports the four service categories (CBR, VBR, ABR, UBR) defined by the ATM Forum and implements explicit rate marking for ABR traffic, backward explicit congestion notification, and a simplified connection admission control. The used formalism has turned out to be quite helpful to develop a readable and scalable model of the switch in a compositional way, despite of the complexity of the switch. Therefore, the experiment we have conducted about the adequacy of the stochastically timed process algebra approach has given a positive outcome, at least from the modeling viewpoint. We have modeled the switch structure only (SES and DES functions are embedded in the inlet/outlet models), because we were interested in evaluating some performance indices of the feedback control mechanism, such as the link utilization, the queue length, and the cell loss ratio. The performance indices we are interested in and the aim of this work allow to resort to some simplifications concerning non ABR mechanisms. When we shall consider end-to-end performance measurements, we shall model each SES (DES) apart (with minimal changes in the present model), and each inlet will support one or more connections. Thanks to these assumptions and the compositional linguistic support of $\mathrm{EMPA}_{\mathrm{vp}}$, we shall be able with a small effort to model more and more complex configurations, with any number of sources, destinations and switches, each one with its feedback scheme (binary or ERM), in order to compute other performance measurements.

From the analysis standpoint, simulation experiments have been conducted in a reasonable time (a few minutes), but the simulation time rapidly increases as the number of sources increases. This can be mitigated by playing with model parameters, because (as explained in Sect. 5) this allows us to reduce the duration of the simulation run obtaining the same results. The same effect can be achieved in this particular framework, where e.g. very low values for the cell loss ratio are expected, by employing rare event simulation techniques in place of ordinary discrete event simulation. In general, we think it is worth investigating to which extent the compositional approach provided by stochastically timed process algebras can be exploited for efficient simulation. We finally recall that performing a simulative analysis on an algebraic specification of a system is more convenient than writing a conventional simulation program, because well known verification techniques can be applied to an algebraic specification to detect functional properties, whereas this is not the case for conventional simulation programs.

We conclude by observing that the actual objective of formal description techniques such as stochastically timed process algebras is not that of being used to assess system properties a posteriori. More specifically, they can be profitably used in the early stages of the design of new systems to predict their performance (as in [12]) in order to avoid cost increases due to the late discovery of inefficiency. This is especially true for stochastically timed process algebras, since they provide a compositional linguistic support which allows system models to be easily modified (usually through changes affecting single terms instead of the whole model) in order to investigate the corresponding variations of performance measures. As far as the ATM switch with ERM is concerned, we plan to suitably make some modifications to the algebraic model presented in this paper in order to predict what improvements can be gained from the performance standpoint.

References

1. M. Ajmone Marsan, *"Stochastic Petri Nets: An Elementary Introduction"*, in *Advances in Petri Nets '89*, LNCS 424:1-29, 1990
2. M. Ajmone Marsan, K. Begain, R. Gaeta, M. Telek, *"GSPN Analysis of ABR in ATM LANs"*, in Proc. of PNPM '97, IEEE-CS-Press pp. 227-236, 1997
3. M. Ajmone Marsan, R. Gaeta, *"Modeling ATM Systems with GSPNs and SWNs"*, in *ACM Performance Evaluation Review*, pp. 28-37, 1998
4. M. Ajmone Marsan, R. Gaeta, *"SWN Analysis and Simulation of Large Knockout ATM Switches"*, in Proc. of ICATPN '98, LNCS 1420:327-344, 1998
5. M. Ajmone Marsan, R. Gaeta, *"GSPN Models of ATM Switches"*, in Proc. of PNPM '97, IEEE-CS-Press pp. 237-246, 1997
6. A. Aldini, M. Bernardo, R. Gorrieri, *"An Algebraic Model for Evaluating the Performance of an ATM Switch with Explicit Rate Marking"*, Tech. Rep. UBLCS-99-11, University of Bologna (Italy), 1999 (`ftp.cs.unibo.it:/pub/TR/UBLCS`)
7. ATM Forum Technical Committee, *"ATM Traffic Management Specification Version 4.0"*, 1996
8. M. Balakrishnan, A. Puliafito, K. S. Trivedi, I. Viniotis, *"Buffer Losses vs. Deadline Violations for ABR Traffic in an ATM Switch: A Computational Approach"*, in Journal of Telecommunication Systems, Design, and Management 7:1-3, 1997
9. M. Bernardo, *"Theory and Application of Extended Markovian Process Algebra"*, Ph.D. Thesis, University of Bologna (Italy), 1999

10. M. Bernardo, W.R. Cleaveland, S.T. Sims, W.J. Stewart, *"TwoTowers: A Tool Integrating Functional and Performance Analysis of Concurrent Systems"*, in Proc. of FORTE/PSTV '98, Kluwer, pp. 457-467, 1998
11. M. Bernardo, L. Donatiello, R. Gorrieri, *"A Formal Approach to the Integration of Performance Aspects in the Modeling and Analysis of Concurrent Systems"*, in Information and Computation 144:83-154, 1998
12. M. Bernardo, R. Gorrieri, M. Roccetti, *"Formal Performance Modelling and Evaluation of an Adaptive Mechanism for Packetised Audio over the Internet"*, in Formal Aspects of Computing 10:313-337, 1999
13. D. Chiu, R. Jain, *"Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks"*, Computer Networks and ISDN Systems 17:1-14, 1989
14. N. Götz, U. Herzog, M. Rettelbach, *"Multiprocessor and Distributed System Design: The Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras"*, in Proc. of PERFORMANCE '93, LNCS 729:121-146, 1993
15. P. Harrison, K. Kanani, *"A Performance Model for SPADES Specifications"*, to appear in Proc. of PERFORMANCE '99, 1999
16. J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996
17. R. Jain, S. Kalyanaraman, R. Goyal, *"Simulation Results for ERICA Switch Algorithm with VBR + ABR Traffic"*, ATM Forum Contribution 95-0467, 1995
18. R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, *"Source Behavior for ATM ABR Traffic Management: An Explanation"*, in IEEE Comm. Magazine, 1996
19. R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, R. Viswanathan, *"ERICA Switch Algorithm: A Complete Description"*, ATM Forum Contribution 96-1172, 1996
20. S. Kalyanaraman, *"Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks"*, Ph.D. Thesis, Ohio State University (OH), 1997
21. L. Kant, W.H. Sanders, *"Performance Analysis of the Knockout Switch under Bursty Traffic based on a Stochastic Activity Network Model"*, in Simulation 70:19-33, 1998
22. G.S. Shedler, *"Generation Methods for Discrete Event Simulation"*, in *"Computer Performance Modeling Handbook"*, S.S. Lavenberg editor, Academic Press, pp. 223-266, 1983
23. K. Siu, R. Jain, *"A Brief Overview of ATM: Protocol Layers, LAN Emulation and Traffic Management"*, in Computer Comm. Review (ACM SIGCOMM), 1995
24. S. Tahar, X. Song, E. Cerny, Z. Zhou, M. Langevin, *"Modeling and Automatic Formal Verification of the Fairisle ATM Switch Fabric using MDGs"*, in IEEE Trans. on CAD of Integrated Circuits and Systems, 1997
25. T. Vassiliou-Gioles, I. Schieferdecker, *"Case Study in Protocol Validation: Validating the ATM Signalling Protocol"*, in Proc. of FMICS '98, 1998
26. P.D. Welch, *"The Statistical Analysis of Simulation Results"*, in *"Computer Performance Modeling Handbook"*, S.S. Lavenberg editor, Academic Press, pp. 267-329, 1983