Analysis of ABR in ATM LANs by GSPN models

M.Ajmone Marsan[1], K.Begain[2], R.Gaeta[3], M.Telek[4]
[1] Politecnico di Torino - Italy
[2] Mu'tah University - Jordan
[3] Università di Torino - Italy
[4] Technical University of Budapest - Hungary

Abstract

This paper discusses the application of the Generalized Stochastic Petri Net (GSPN) modeling paradigm to the performance analysis of Asynchronous Transfer Mode (ATM) networks. GSPNs are first shown to be an adequate tool for the development of models of ATM networks, provided that only one timed transition is used, together with many immediate transitions. The only timed transition in the GSPN represents the ATM network cell time, while immediate transitions implement the ATM network behavior. The firing time distribution of the only timed transition in the GSPN is irrelevant for the computation of several interesting performance indices. The development of GSPN models of an ATM LAN with users exploiting the UBR (Unspecified Bit Rate) and ABR (Available Bit Rate) ATM service categories demonstrates the flexibility of the approach.

1 Introduction

Performance problems are quite relevant in Asynchronous Transfer Mode (ATM) networks, whose objective is to provide different services with suitable Quality of Service (QoS) guarantees. Performance problems in ATM are quite difficult because of the complexity of the systems, and require investigation at many different levels of abstraction. When the considered level of abstraction focuses on the cell-scale dynamics (e.g. for the computation of cell loss probabilities due to buffer overflows), the discrete-time nature of the cell transmission process sometimes makes the continuous-time modeling approaches difficult to apply. Nevertheless, continuous-time models continue being used and adapted to the ATM environment.

In this paper we describe an approach for the adaptation of the Generalized Stochastic Petri Net (GSPN) modeling paradigm [1, 2] to the performance analysis of ATM networks at the cell level, and illustrate its use by means of an example, providing both numerical performance results and indications about the complexity of the proposed approach.

Few previous examples of the use of PN-based approaches for the performance analysis of ATM networks exist in the literature [3, 4, 5]. The first paper [3] uses a GSPN approach to model and evaluate cell scheduling policies in ATM multiplexers. In [4] the authors develop a GSPN model of the Gauss ATM switch

[6] using a structural decomposition of their model to circumvent the state space explosion problem, and approximate the deterministic cell transmission times with Erlang-10 distributions. In [5], the authors develop a model of the Knockout ATM switch [7] using Stochastic Activity Networks (SANs) [8], adopting an approach very similar to the one that will be described below for the representation of the deterministic cell transmission times.

2 The GSPN Approach to model ATM Networks

What type of Petri Net (PN) models can be adequate for the description and analysis of the dynamics of ATM systems? The obvious answer to this question is that we should like to use a PN modeling paradigm in which we can freely mix transitions with constant delays (both null and non-null), and transitions with exponentially distributed random delays. The latter are useful for the description of the user behaviors that result in the network workload; the former are necessary for the description of the internal behavior of the ATM network subsystems. This leads to requirements that come quite close to the definition of DSPNs [9, 10]. Indeed, with the DSPN modeling paradigm it is possible to use deterministic, immediate, and exponential transitions within one model. Unfortunately, the use of deterministic transitions within DSPNs is limited by quite stringent conditions for the resulting stochastic process to be studied with acceptable complexity (which essentially boil down to the fact that just one deterministic transition can be enabled at any given time instant), and even under those restrictions the computational cost for the numerical study of medium and large DSPN models is exceedingly high.

However, considering in more detail the cell-level dynamics of an ATM network, we can realize that the generality that we required for the use of deterministic transitions is actually not necessary; indeed the data rates (and hence the cell rates) on the transmission channels of ATM networks often are either equal, or integer multiples of a common value (for example, long distance ATM networks normally use bit rates close to 155.5, 622, and 1244 Mb/s). Moreover, also the time slots used inside ATM equipments are tied to the cell rates on the transmission channels, so that it is generally true that in a model of an ATM network or of one of its subsystems, the deterministic delays are either all equal, or integer multiples (according to small integer constants) of one deterministic time. We can thus imagine that in most cases it can be possible to construct a model in which one deterministic transition is sufficient to provide a sort of clock that drives all events that last for deterministic times within the considered system. This brings us back to DSPNs, but in this case we need only one deterministic transition in the model to implement the clock, and the only issue that remains open concerns the computational cost of the solution.

Furthermore, if we accept some reduction in the flexibility in the description of the user behavior, and thus of the network workload, we can also eliminate all exponential transitions, replacing them with immediate transitions whose weights are set so as to more abstractly model the user behavior. This leaves us with a PN model in which all transitions are immediate, with just one exception: one transition is timed with a constant delay τ defining the time unit in our model. Note that this timed transition actually defines the clock of our model and thus always has concession. The stochastic process generated by the dynamic behavior of such a PN model is a semi-Markov process (SMP) with constant sojourn times, with an embedded discrete-time Markov chain (DTMC), whose evolution over the state space is isomorphic to the tangible marking process, and whose transition probabilities are computed from the reachable markings and from the weights of the enabled immediate transitions.

However, the association of the only timed transition with either a constant or an exponentially distributed random delay makes no difference for the computation of a large quantity of interesting performance parameters. Indeed, while the PN model with the deterministic transition originates a DTMC, the PN model with the exponential transition originates a continuous-time Markov chain (CTMC); the relation between the two MCs is very tight: the DTMC is the embedded MC of the CTMC. It is well known that (see e.g. [2]) the steady-state probabilities $\pi_k^{(Y)}$ of an ergodic CTMC $Y(t)$ can be obtained from the stationary probabilities $\pi_k^{(Z)}$ of its embedded DTMC Z_n through the relation $\pi_k^{(Y)} = \frac{\pi_k^{(Z)} E[W_k]^{(Y)}}{\sum_{i \in S} \pi_i^{(Z)} E[W_i]^{(Y)}}$ where S is the state space of the two MCs, and $E[W_i]^{(Y)}$ is the average sojourn time in state i for the CTMC $Y(t)$. Since $\forall k \in S : E[W_k]^{(Y)} = \tau$ (throughout the paper we shall assume $\tau = 1$), the steady-state probabilities $\pi_k^{(Y)}$ are identical to the stationary probabilities $\pi_k^{(Z)}$ of the embedded MC. We can thus compute the performance of ATM networks by developing appropriate GSPN models in which only one transition is timed, and all other transitions are immediate.

Note that the approach we just outlined for the description of deterministic times within GSPN models is not novel: a similar procedure was first adopted in some unpublished works by G.Balbo for the investigation of multiprocessor systems, and the application of this approach to PN models of ATM networks was for the first time proposed in [5], considering SAN rather than GSPN models. Actually, our proposal is a slight generalization of the approach in [5], since we consider one clock from which different deterministic times can be constructed as multiples of the clock period; this allows considering different link rates, as well as the internal speed-up of the ATM network components. In [5], instead, only one deterministic cell time is considered.

3 The ATM LAN

As an example of the use of GSPNs to model ATM networks, we consider an ATM LAN of the type depicted in Fig. 1, where users are assumed to exploit either the UBR or the ABR ATM service category. The UBR (Unspecified Bit Rate) ATM service category does not require sources to control their cell transmission rates. Instead, the ABR (Available Bit Rate) ATM service category requires sources to adapt their cell rate to the network indications.

The ATM switches in the LAN are assumed to be non-blocking with one finite buffer for the storage of cells to be transmitted on each output link. The maximum number of cells that can arrive in the same slot at one output buffer equals the number of switch input channels (say N). If the free space in the output buffer is not sufficient to store all cells that arrive in a slot, some cells are lost.

Note that we take an abstract view of the switch behavior; the switch speed-up is not mentioned, and no cell losses within the switching fabric are considered. Thus, the ATM switch is essentially modeled by the cell queues at output ports.

Flow control in ABR is based on *Resource Management* (RM) cells that are periodically inserted within the flow of data cells along the connection; these RM cells travel from source to destination (forward RM cells), and then return to the source (backward RM cells). The ATM switches along the connection can use RM cells to control the rate at which ABR sources inject cells into the network, in order to both efficiently exploit the available bandwidth and prevent congestion.

A simple algorithm suggested by the ATM forum [11] for the control of ABR connections is called Relative Rate Marking (RRM), and is based on the *Congestion Indication* (CI) and *No Increase* (NI) bits contained in the RM cells, whose values can be combined in a three-state feedback corresponding to the indications "increase rate", "keep rate", or "decrease rate". Depending on the received feedback, sources change their transmission rates according to negotiated parameters called *Rate Increase Factor* (RIF) and *Rate Decrease Factor* (RDF).

The feedback generation is governed by the congestion control algorithm of the switch, that often is based on the occupancy of the output link buffer. Two thresholds are normally defined for the buffer. Congestion is detected when the buffer occupancy surpasses the high threshold, while underutilization of resources is assumed when buffer occupancy is below the low threshold.

The non-blocking characteristics of the ATM switches in the LAN, and the nature of ABR flow control, are such that the QoS perceived by an ABR end user is mainly determined by the operations of the most congested ATM switch traversed by the ABR ATM connection. For this reason, we shall not model the whole ATM LAN (which is too complex to be described with one GSPN model), but focus on the congested ATM switch, and on all traffic flows that inject cells into the switch output buffer used by the ABR connection. A system model of the ATM LAN can thus become as shown by the part of Fig. 1 drawn with thick lines, where we assume that two UBR and one ABR user interact with a server, and congest the switch output buffer that stores cells directed to the server; this setup will be modeled with GSPNs.

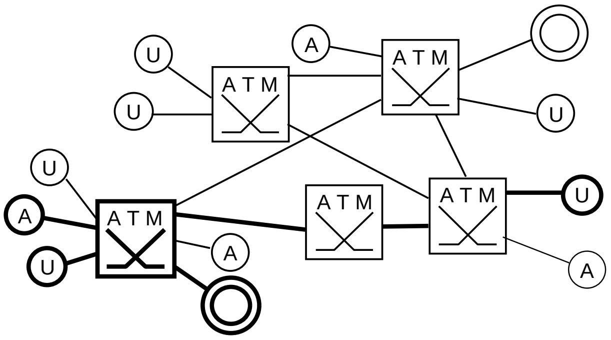

Figure 1: The ATM LAN; ABR and UBR users are marked A and U, respectively; servers are depicted as double circles. The elements that are described within the GSPN model are drawn with thick lines.

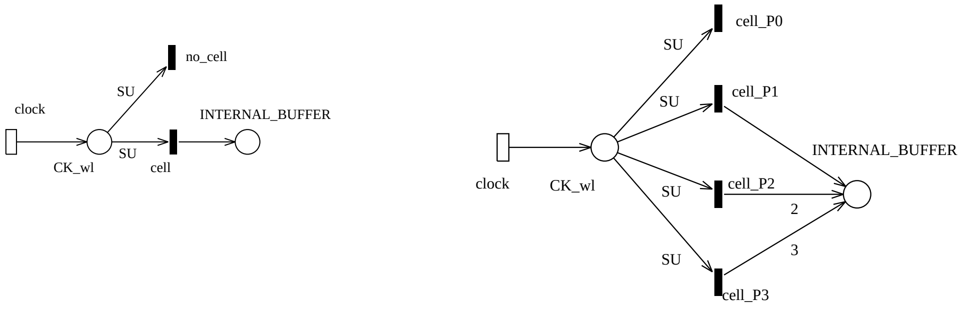

Figure 2: Bernoulli workload models

4 GSPN Model of the ATM LAN

The GSPN description of the model of the ATM LAN can be constructed with a modular approach, separately describing the behaviors of the following components:

- the cell generation processes due to UBR sources (the *UBR source models*)
- the cell generation process due to the ABR source (the *ABR source model*)
- the queue at the output port (the *output model*)
- the ABR feedback (the *feedback model*)

These model components will be described separately. We shall then combine these components to study the behavior of a LAN with one ABR source.

The GSPN model components comprise only immediate transitions, whose firing is driven by the only timed transition in the model, that is named *clock*, and whose firing delay represents the cell time in the LAN.

When *clock* fires, one token is deposited in the places named CK_wl, CK_wl_abr, CK_out, and CK_delay in the following subsections. The presence of tokens in these places activates the immediate transitions within GSPN model components.

4.1 The UBR Source Models

Similarly to what happens in most commercial ATM switches, where the input cell flows are synchronized before entering the switching fabric, we assume that the arrivals of cells at input ports are synchronized. Thus, at each slot any source can either generate zero or one cell, according to the chosen source behavior.

4.1.1 Bernoulli source models

The GSPN model for one UBR source producing a Bernoulli cell flow is depicted in the left part of Fig. 2. One token in place CK_wl enables the two conflicting immediate transitions *cell* and *no_cell*. The firing of *no_cell* (whose weight is $1-p$) indicates no cell arrival at the input port during the current time slot, whereas the firing of *cell* (whose weight is p) models the arrival of a cell. This cell is transferred to the internal buffer, modeled by place SWITCHED_CELLS.

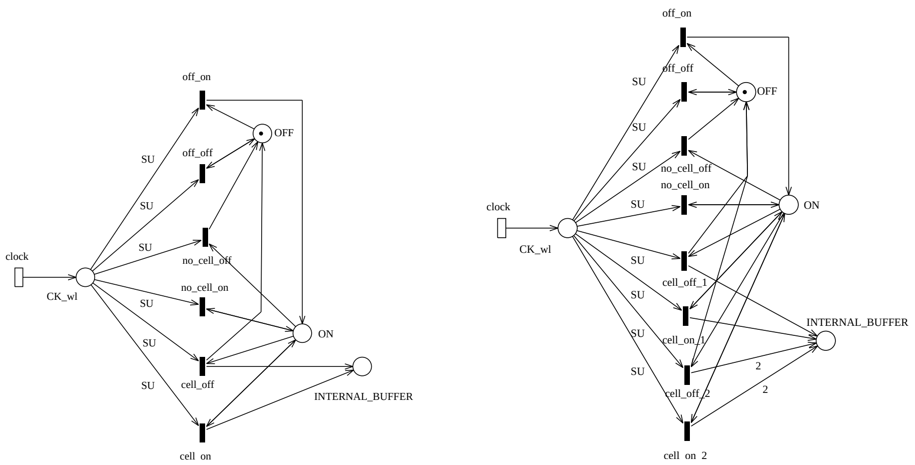

Figure 3: MMBP workload models

When more than one UBR Bernoulli sources are connected to the switch input ports, whether they all have the same parameter p, or different parameters p_j, it is possible to provide an aggregate description rather than a brute force replication of models like the one we just discussed. This obviously implies a reduction in the number of states of the GSPN model.

Aggregate descriptions of UBR Bernoulli sources allow the arrival of more than one cell within one time slot, and must thus be connected to switching fabric models that accept as inputs the superpositions of several input cell flows.

The aggregate GSPN model of K Bernoulli sources comprises $K+1$ transitions *cell_Pi*, with $i = 0, 1, \cdots, K$ that generate $0, 1, \cdots, K$ tokens in place SWITCHED_CELLS. The weight of the ith transition (the one that generates i tokens) is easily calculated as $w_i = \left(\begin{array}{c} K \\ i \end{array} \right) p^i (1-p)^{K-i}$ if all UBR Bernoulli sources have the same parameter p. Instead, when UBR Bernoulli sources have different parameters p_j, the weight of the ith transition becomes

$$w_i = \sum_{\forall \boldsymbol{k}_i} \left[\prod_{\forall k_n = 1} p_n \prod_{\forall k_m = 0} (1 - p_m) \right]$$

where the $\boldsymbol{k}_i$ are all possible vectors with i entries k_n equal to 1, and $K - i$ entries k_m equal to 0. As an example, the right part of Fig. 2 shows this aggregate model for three Bernoulli sources.

4.1.2 MMBP and MMDP source models

Slightly more elaborate GSPN workload models can account for more complex cell arrival processes. For example, Markov-modulated source models can be simply described with GSPNs; this is a class of workload models that is commonly used when studying ATM networks, since it provides larger correlations in the cell arrival streams, thus better approximating the behavior of real users. In the left part of Fig. 3 we depict the workload model for one source generating a cell flow that is the discrete-time version of a Markov-modulated Poisson process (a Markov Modulated Bernoulli Process – MMBP).

In this case the arrival process can be either on (place ON is marked) or off (place OFF is marked). If the arrival process is on, the four transitions *cell_on*, *cell_off*, *no_cell_on*, and *no_cell_off* are enabled if one token is in place CK_wl. The firing of one of the two transitions *cell_on* and *cell_off* models the arrival of a cell that is transferred to the internal buffer modeled by place SWITCHED_CELLS. Transition *cell_on* (whose weight is $pP(on - on)$) leaves place ON marked, whereas transition *cell_off* (whose weight is $p[1 - P(on - on)]$) removes the token from place ON and deposits a token in place OFF, thus modeling the state of the arrival process in the next slot. Similarly, the firing of one of the two transitions *no_cell_on* and *no_cell_off* (with weights $(1-p)P(on-on)$ and $(1-p)[1-P(on-on)]$, respectively) models the lack of a cell arrival during the current slot, and the state of the arrival process in the next slot.

If the arrival process is off, no cell can arrive: the two transitions *off_on*, and *off_off* (with weights $1 - P(off - off)$ and $P(off - off)$, respectively) are enabled if one token is in place CK_wl, and the firing of one of the two yields the state of the arrival process in the next slot.

With this MMBP source model, the cell arrival process on and off periods are geometrically distributed random variables, whose averages are the inverses of the probabilities $1 - P(on - on)$ and $1 - P(off - off)$. The source *activity factor* (AF) is defined to be ratio between the average on period duration and the sum of the average on and off period durations; with trivial algebra we get:

$$AF = \frac{1 - P(off - off)}{[1 - P(on - on)] + [1 - P(off - off)]}$$

The *average load* generated by the MMBP source is $\rho = p\ AF$ and the *average burst size* is $BS = p/(1 - P(on - on))$. The cell interarrival times during the on periods are geometrically distributed random variables, with average $1/(1 - p)$.

A further minor modification of the GSPN workload model can lead to the representation of cell arrival streams following a Markov-modulated Deterministic Process (MMDP), which also is often used in the study of ATM systems. In the case of MMDP sources, the arrival process can be either on or off (like for MMBP sources), but, when the process is on, one cell surely arrives in each slot. The GSPN model for a MMDP source is obtained from that depicted in the left part of Fig. 3 simply by deleting the two transitions *no_cell_on* and *no_cell_off*.

Also in this case, compact representations of groups of MMBP or MMDP sources loading the same input are easy to build; the aggregate description of sources is specially compact when the modulating process is the same for all sources in the group. As an example of compact representation of a group of MMBP sources, we show in the right part of Fig. 3 the aggregate workload model for two MMBP sources with the same modulating process.

4.2 The ABR source model

The key feature of an ABR source is its ability to adapt its cell generation rate according to the feedback arriving from the network within RM cells. In real

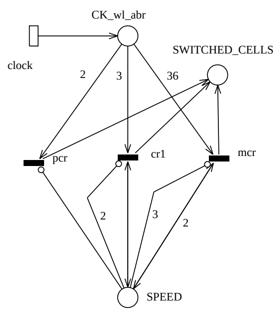

Figure 4: ABR source model

systems, the possible cell rates of an ABR source are very numerous, but finite. In our approximate GSPN model, the ABR source can select its cell rate within a small set of values only. In particular, we develop models where the ABR source cell rate can be equal to a subset of the integer divisors of the link cell rate ($C/2$, $C/3$, $C/4$, etc., where C is the link cell rate). We denote the number of speeds of an ABR source as ns.

Fig. 4 shows the ABR source model in the particular case in which the possible cell rates relative to the link speed are $1/2$, $1/3$, and $1/36$. This means that the peak cell rate (PCR) of the ABR source is $C/2$, and that the minimum cell rate (MCR) is $C/36$. The marking of place SPEED determines the cell rate: depending on the number of tokens in place SPEED, when a token is generated in place CK_wl_abr, one of the three immediate transitions pcr, mcr, $cr1$, may become enabled.

Transition pcr is enabled if place SPEED is empty, and place CK_wl_abr contains two tokens; its firing removes two tokens from place CK_wl_abr, and generates one token in place SWITCHED_CELLS. The marking of place SPEED is not altered. The firing of transition pcr thus models the generation of a cell every other clock time, hence the ABR source transmission at its PCR equal to half the link cell rate.

Transition mcr is enabled if place SPEED contains two tokens, and place CK_wl_abr contains 36 tokens; its firing removes 36 tokens from place CK_wl_abr, and generates one token in place SWITCHED_CELLS. The marking of place SPEED is not altered. The firing of mcr thus models the generation of a cell at every thirty-sixth clock time, hence the ABR source transmission at its MCR equal to $C/36$.

Similarly, transition $cr1$ models the generation of a cell at every third clock time, hence the ABR source transmission at one third of the link rate.

Of course, using this approach it is trivial to add transitions modeling speeds equal to one n-th of the link cell rate. However, this results in more complex behaviors, hence in increased reachability set sizes. Fortunately, the increase in the reachability set size is linear with the number of ABR source speeds.

4.3 The output model

The output port buffer stores the cells that await their turn for transmission on the output channel, and a transmitter that can load a cell onto every slot available on the output link.

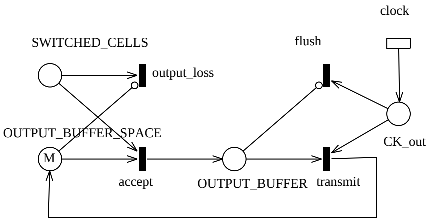

Figure 5: Output interface models

The GSPN model of the output port queue is shown in Fig. 5. The number of tokens in place OUTPUT_BUFFER_SPACE, initially set to the value M, indicates the free space in the output buffer, whereas the number of tokens in place OUTPUT_BUFFER indicates the number of cells awaiting transmission.

Cells that arrive from the switching fabric are modeled by tokens in place SWITCHED_CELLS. When a cell arrives from the switching fabric, either one of the two transitions *output_loss* or *accept* can be enabled. The firing of *output_loss* models the loss of a cell due to the lack of space in the output buffer (no token is present in OUTPUT_BUFFER_SPACE). The firing of *accept* instead models the acceptance of the cell into the output buffer; one token is removed from OUTPUT_BUFFER_SPACE, and one token is generated in OUTPUT_BUFFER.

At each firing of *clock*, one token is generated into place CK_out; if one cell exists in the output buffer (at least one token is in place OUTPUT_BUFFER), the transmission is completed through the firing of transition *transmit*, removing one token from place OUTPUT_BUFFER and adding one position to the free output buffer spaces (OUTPUT_BUFFER_SPACE). Otherwise, the token in place CK_out is flushed.

4.4 The ABR feedback model

ATM switches implementing the RRM ABR scheme often determine their congestion status depending on buffer occupancy, and issue feedback to ABR sources accordingly. As mentioned before, in our model the number of tokens in place SPEED determines the cell rate of the ABR source. Thus, the feedback generated by the ATM switch towards the ABR source is reflected in the modification of the marking of place SPEED.

Fig. 6 shows the GSPN model for the RRM ABR feedback mechanism. The model comprises two main parts: 1) the measurement of the buffer occupancy and the decision about the type of feedback to be returned to the source, and 2) the propagation of the feedback to the source and the modification of the source state.

Immediate transitions *ir_is_sent*, *ir_is_not_sent*, *kr*, *dr_is_sent*, and *dr_is_not_sent*, in Fig. 6 model the measurement of the buffer occupancy and the decision about the type of feedback to be returned to the source: Increase Rate, Keep Rate, and Decrease Rate, respectively. The measurement and the feedback decision are implemented every Dm slots, i.e., whenever Dm tokens accumulate in place CK_delay. This measurement delay Dm models the cadence of RM cells within the flows of cells that traverse the switch (according to [11], each source inserts one RM cell every Nrm data cells, and we choose $Dm = Nrm$).

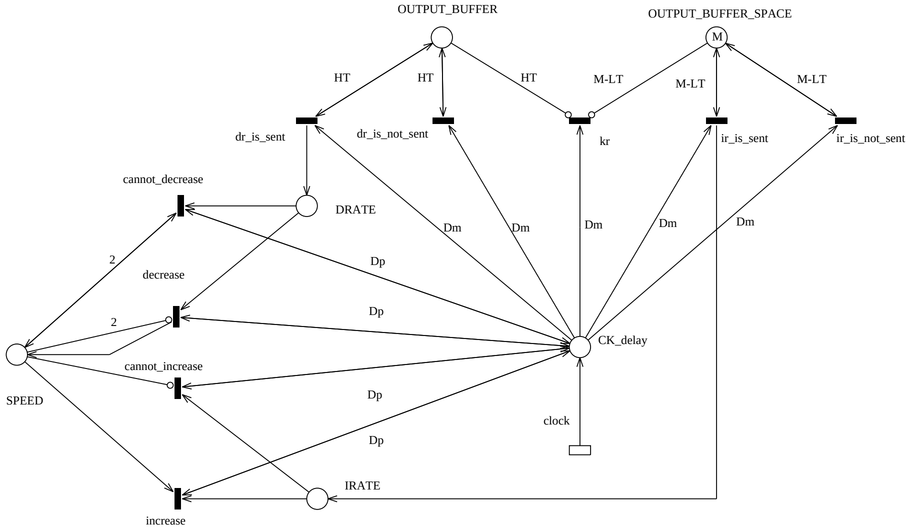

Figure 6: ABR feedback model

Transitions *dr_is_sent* and *dr_is_not_sent* correspond to a high congestion situation, and are enabled when the buffer occupancy exceeds the high threshold HT; this is modeled by the test arcs with multiplicity HT from place OUTPUT_BUFFER to the two transitions. Transitions *dr_is_sent*, and *dr_is_not_sent* form a free-choice conflict, whose probabilistic resolution models the availability of a RM cell to carry the feedback within Dm slots. Only if transition *dr_is_sent* fires, one token is generated into place DRATE. When the ABR source is transmitting at cell rate C/n, the weight of transitions *dr_is_sent* is set to $1/n$, and that of transitions *dr_is_not_sent* is set to $1 - 1/n$. This is because the source transmits one RM cells every $n \times Nrm$ slots, and since $Dm = Nrm$, the probability of finding an RM cell in a sequence of Dm slots is $1/n$.

Note that firing either transition does not alter the marking of place OUTPUT_BUFFER, and removes the Dm tokens from place CK_delay.

Transitions *ir_is_sent* and *ir_is_not_sent*, correspond to an underutilization situation, and are enabled when the buffer occupancy is below the low threshold LT; in this case the multiplicity of the test arcs from place OUTPUT_BUFFER_SPACE to the two transitions is $M - LT$, where M is the buffer size, and LT is the low threshold. As before, transitions *ir_is_sent*, and *ir_is_not_sent* form a free-choice conflict, whose probabilistic resolution models the availability of a RM cell to carry the feedback in Dm slots. Only if transition *ir_is_sent* fires, one token is generated into place IRATE. Similarly to the previous case, the weight of transition *ir_is_sent* is set to $1/n$, and that of transition *ir_is_not_sent* is set to $1 - 1/n$ when the ABR source is transmitting at cell rate C/n.

Again, firing either transition does not alter the marking of place OUTPUT_BUFFER, and removes the Dm tokens from place CK_delay.

If the buffer occupancy is between LT and HT, transition *kr* is enabled, and its firing only removes the Dm tokens accumulated in place CK_delay, without producing any explicit feedback. This is equivalent to the generation of either a Keep Rate feedback, or no feedback, since the ABR source cell rate in this case need not be changed.

The second phase of the ABR feedback control consists in the propagation of the feedback to the ABR source and in the modification of the source state. This is assumed to require a constant delay, that equals the propagation delay of RM cells from the switch to the ABR source. For this reason the modification of the ABR source state takes place not immediately after the feedback generation, but after Dp slots, that is after Dp tokens accumulate in place CK_delay.

If the first phase resulted in the generation of one token in place DRATE, then one of the two mutually exclusive transitions *decrease* and *cannot_decrease* is enabled, and its firing will remove the token from place DRATE. Only if the marking of place SPEED is less than 2, a token will be generated in it (firing of *decrease*), in order to decrement the cell rate of the ABR source; otherwise, the marking of place SPEED remains unchanged (firing of *cannot_decrease*).

If instead the first phase ended with one token in place IRATE, then one of the two mutually exclusive transitions *increase* and *cannot_increase* is enabled, and its firing will remove the token from place IRATE. Only if the marking of place SPEED is nonzero, a token will be removed from it (firing of *increase*), in order to increment the cell rate of the ABR source; otherwise, the marking of place SPEED remains unchanged (firing of *cannot_increase*).

Obviously, transitions *decrease*, *cannot_decrease*, *increase*, and *cannot_increase* must have higher priority than transitions *ir_is_sent*, *ir_is_not_sent*, *kr*, *dr_is_sent*, and *dr_is_not_sent*, so that the second phase of the previous control cycle can be finished before starting the next cycle.

In order to conclude the description of the ABR feedback model, two observations are necessary. First, note that we assumed a constant delay both between two consecutive measurements of the buffer occupancy (Dm slots), and between the feedback generation and its interpretation at the ABR source (Dp slots). These two delays need not be equal, but for our model to work properly it is necessary that $Dp \leq Dm$. If this is not the case, a simple modification allows considering longer propagation delays (at the expense of an increase in the reachability set cardinality). Second, note that when the ABR source speed is incremented, say from C/n_1 to C/n_2, with $n_1 > n_2$, in general it might happen that place CK_wl_abr (see Fig. 4) contains a number of tokens larger than kn_2, with $k \geq 2$, so that the model would generate k cells within one slot time. This behavior is obviously incorrect, and must thus be avoided. One possibility to exclude this behavior is through adequate changes in the GSPN model; another possibility is by setting Dm, the interval between two consecutive updates of the marking of place CK_wl_abr to be a multiple of all divisors n_i of the link speed C that result in the allowed rates of the ABR source C/n_i. The latter option will be used in this paper.

4.5 The Complete GSPN Model

The complete GSPN model of the portion of the ATM LAN that refers to one of the switch output channels can be constructed by assembling the modules that were just described.

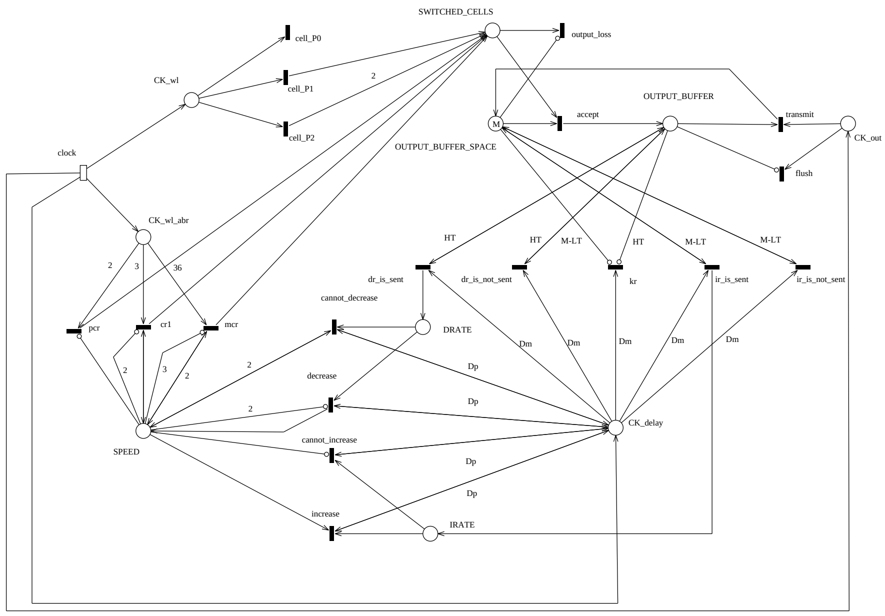

Figure 7: Complete GSPN model of the ATM LAN with one ABR source with three speeds

As an example, the complete model of an ATM LAN where the congested switch has a given number of input (and output) ports, 2 of which are loaded with UBR traffic destined to the considered output, while another input is used by an ABR source that also injects traffic towards the considered output, is shown in Fig. 7.

It can be immediately observed that this model results from the composition of the submodels that were previously described; we therefore omit its description.

The assignment of priorities to the different immediate transitions in the model is quite a delicate matter, since it defines the sequence of operations that are performed in zero time after the firing of *clock*. Only a careful setting of the priorities results in a correct model.

The priorities of all transitions in the model are presented in Table 1. Pri-

Table 1: Immediate transitions priority for the ATM LAN GSPN model

Transition name	Priority value
kr, *ir_is_sent*, *ir_is_not_sent*	1
dr_is_sent, *dr_is_not_sent*	1
decrease, *cannot_decrease*	2
increase, *cannot_increase*	2
$cell\_P_i$, *pcr*, *mcr*, *cr*1	3
output_loss, *accept*	4
transmit, *flush*	5

orities in our models increase from the cell source section to the switch output interface section, since they must avoid that a cell that was just generated crosses the switching fabric, and the switch output interface in zero time. Assigning the highest priority to transitions in the output interface model allows the management of the transmission side first, before new cells cross the switch. The second set of operations (modeled by intermediate priority transitions) moves cells to the output buffer while the last set of operations concerns the source and the ABR feedback mechanism models, which comprise transitions at the lowest priority levels.

5 Numerical Results

In this section we present and discuss numerical results obtained from the solution of the GSPN models of the ATM LAN with one ABR user, and we compare the GSPN results with those obtained from very detailed simulation models. The simulation results were obtained with CLASS [12], a software tool for the simulation of ATM networks at the cell level developed by Politecnico di Torino, in cooperation with CSELT, the research center of Telecom Italia. CLASS carefully implements all the details of the source, destination, and switch behaviors described in the ABR specification [11].

Results were obtained from GSPN models with the *GreatSPN* package [13] on a Sun SPARC 10 workstation equipped with 80 MBytes of main memory and running SunOS 4.1.3.

In the following $X(t)$ denotes the throughput of immediate transition t. We focus on three performance indexes:

- *Output Link Utilization* (OLU), defined as $OLU = X(transmit)$
- *Average ABR Source Rate* (AAR), defined as $AAR = X(pcr) + X(mcr) + \sum_{j=1}^{ns-2} X(cr\_j)$. Note that also $\sum_{i=1}^{na} AAR_i = \rho_a$ where na is the number of ABR sources and ρ_a is the load offered to the output link by ABR sources
- *Cell Loss Ratio* (CLR), defined as $CLR = \frac{X(output\_loss)}{\rho}$ where $\rho = \rho_a + \rho_u$ is the total offered load to the output link, obtained as the sum of the loads of ABR and UBR sources, with $\rho_u = \sum_{i=0}^{nu} i \cdot X(cell\_P_i)$ where nu is the number of UBR sources

The curves referring to the cell loss ratio are plotted using a logarithmic scale for the vertical axis.

We consider a small ATM LAN where the buffer storing cells directed to the considered output port of the congested switch receives from 2 input ports UBR traffic (either Bernoulli or MMBP), as well as the ABR traffic generated by one ABR source.

The first set of numerical results is presented in the six graphs of Fig. 8, that show the cell loss ratio (top), the output link utilization (middle), and the average ABR source rate (bottom), as functions of the output buffer size, in the case of either Bernoulli (left) or MMBP (right) UBR traffic. The parameter values used to derive this first set of numerical results are reported in Table 2.

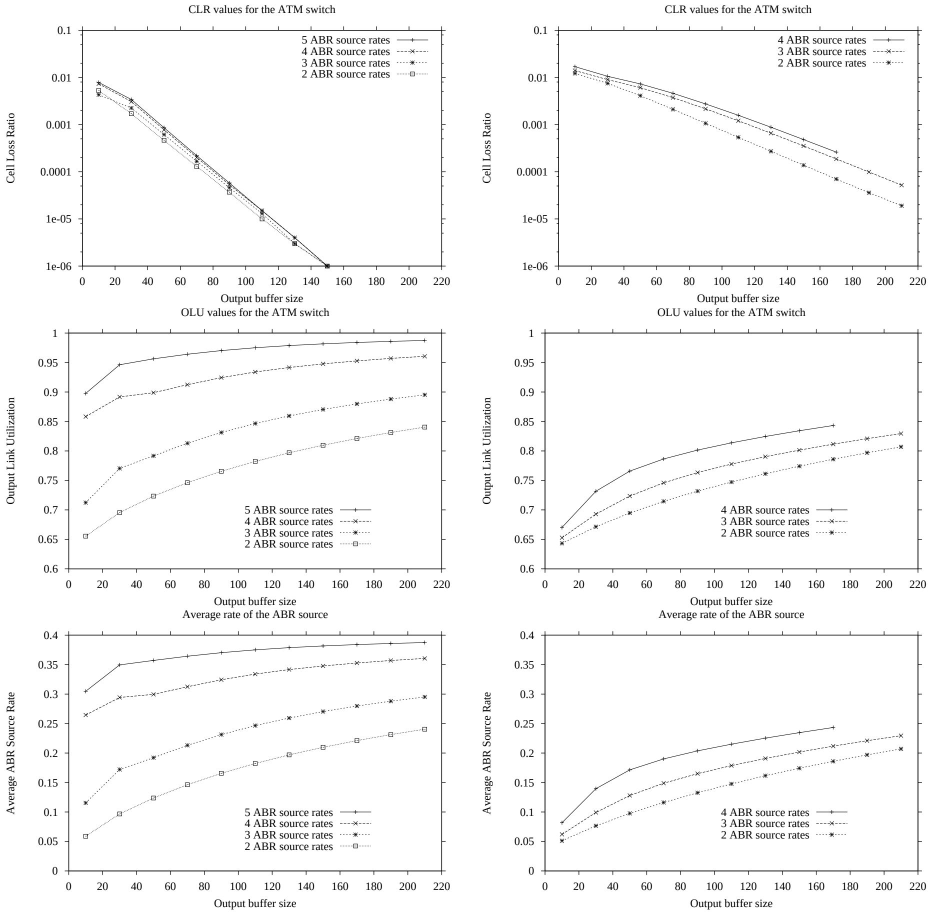

Figure 8: Cell loss ratio (top), output link utilization (middle), and average ABR source rate (bottom), as functions of the output buffer size, for either Bernoulli (left) or MMBP (right) UBR traffic

The round trip propagation delay from the ABR source to the ATM switch is 18 cell times. Translating this value into a distance depends on the considered link cell rate. Indeed, in a high data rate LAN running at 155 Mbit/s the cell transmission time is about 2.7 μs, and thus corresponds to about half a km. Instead, in a lower data rate LAN running at 25 Mbit/s (a standard speed for ATM LANs) the cell transmission time is about 17 μs, and thus corresponds to about 3 km. In the former case the selected propagation delay value corresponds approximately to a distance equal to 4.5 km; in the latter case to a distance equal to 27 km.

The MMBP model used for the description of the two UBR sources that interfere with the only ABR source is rather pessimistic: the correlation between the two UBR sources is maximized by using the same modulating process for both, and by assuming that cell generations from the two sources always coincide. The latter assumption can be quite easily removed with no impact on the size of the tangible reachability set.

From the graphs of Fig. 8 we first of all note that the MMBP UBR traffic induces

Table 2: Parameters of the ATM LAN used in the derivation of the first set of numerical results; C is the link cell rate

Parameter	Value
measurement delay	$Dm = 36$
propagation delay	$Dp = 18$
number of ABR sources	1
number of ABR source rates	2 or 3 or 4 or 5
ABR source PCR	$C/2$
ABR source MCR	$C/36$
ABR source additional rates	$C/3$, $C/4$, $C/6$
number of UBR input ports	2
UBR traffic type	Bernoulli or MMBP
UBR traffic load	$\rho_u = 0.6\ C$
MMBP source activity factor	$AF = 0.8$
MMBP source burst size	$BS = 128$
output buffer size	M (variable)
low buffer threshold	$0.1\ M$
high buffer threshold	$0.6\ M$

remarkably worse performance than Bernoulli UBR traffic for all the considered indexes: the average ABR source rate remains quite far from the value $0.4\ C$ that is available after the UBR load is subtracted from the link capacity, even with a larger number of ABR speeds (while with Bernoulli UBR traffic and 5 ABR speeds, the curves almost reach the target); the output link utilization hardly exceeds 0.8 (while with Bernoulli UBR traffic and 5 ABR speeds, the curves almost reach 0.98); the cell loss ratio never goes below 10^{-5} (while with Bernoulli UBR traffic, cell loss ratios decrease below the precision of the numerical analysis, set to 10^{-6}).

Second, we can observe that a larger number of speeds at the ABR source allows a better utilization of the output link, due to a higher average ABR source rate. The difference is quite large in the case of Bernoulli UBR traffic, but much less for MMBP UBR traffic. Of course, the reduced utilization of the output link with just two speeds decreases the cell loss ratio, specially with MMBP UBR traffic.

The second set of numerical results is presented in the six graphs of Fig. 9, that show the same performance indexes as in the previous figure, now as functions of the UBR traffic load ρ_u, in the case of either Bernoulli (left) or MMBP (right) UBR traffic.

The parameter values used in the derivation of this second set of numerical results are as in Table 2, except for those listed in Table 3.

The graphs of Fig. 9 again show that the MMBP UBR traffic yields much worse performance than Bernoulli UBR traffic for all the considered indexes: the average ABR source rate, that cannot be larger than its PCR value $0.5\ C$, starts decreasing at $\rho_u = 0.5$ with Bernoulli UBR traffic, and at $\rho_u = 0.3$ with MMBP UBR traffic; the output link utilization never approaches 0.9 with MMBP UBR traffic, while with Bernoulli UBR traffic and 5 ABR speeds the curve remains between 0.9 and 1

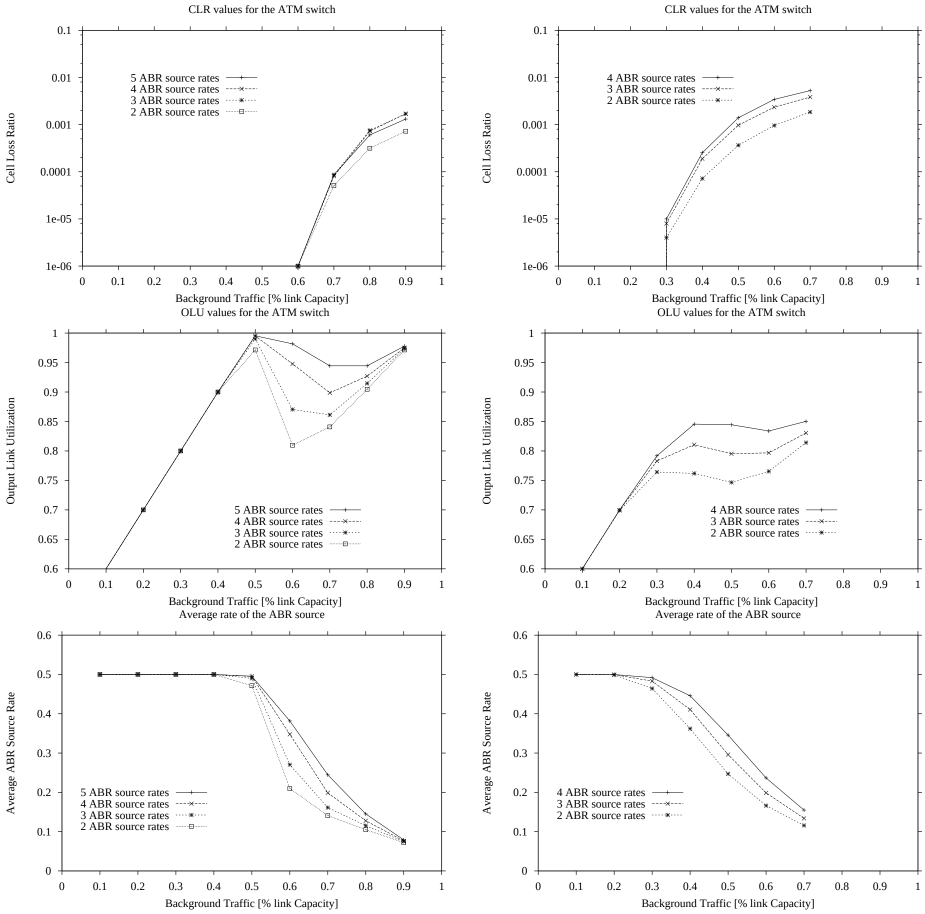

Figure 9: Cell loss ratio (top), output link utilization (middle), and average ABR source rate (bottom), as functions of the UBR traffic load, for either Bernoulli (left) or MMBP (right) UBR traffic

for UBR traffic loads above 0.5; the cell loss ratio grows much later with Bernoulli UBR traffic than with MMBP UBR traffic.

Also in this case we can observe that a larger number of speeds at the ABR source has a beneficial impact on the LAN performance.

The difference in performance observed with Bernoulli and MMBP UBR traffics deserves some deeper investigation: in particular it can be interesting to study how the average burst size impacts the position of the curves that were just discussed for MMBP UBR traffic.

Results for variable average burst size in the case of MMBP UBR traffic are reported in Fig. 10, in the same conditions of Fig. 9, for the case of 2 ABR source rates (left column) and 4 ABR source rates (right column).

As expected, longer on periods (that translate in higher correlations in the UBR traffic) have a dramatic impact on the LAN performance, for all the considered indexes.

In order to validate the behavior of the GSPN models in the case of ABR sources

Table 3: Parameters of the ATM LAN used in the derivation of the second set of numerical results

Parameter	Value
UBR traffic load	ρ_u (variable)
MMBP source activity factor	$AF = \rho_u + 0.2$
MMBP source burst size	$BS = 16$
output buffer size	$M = 150$

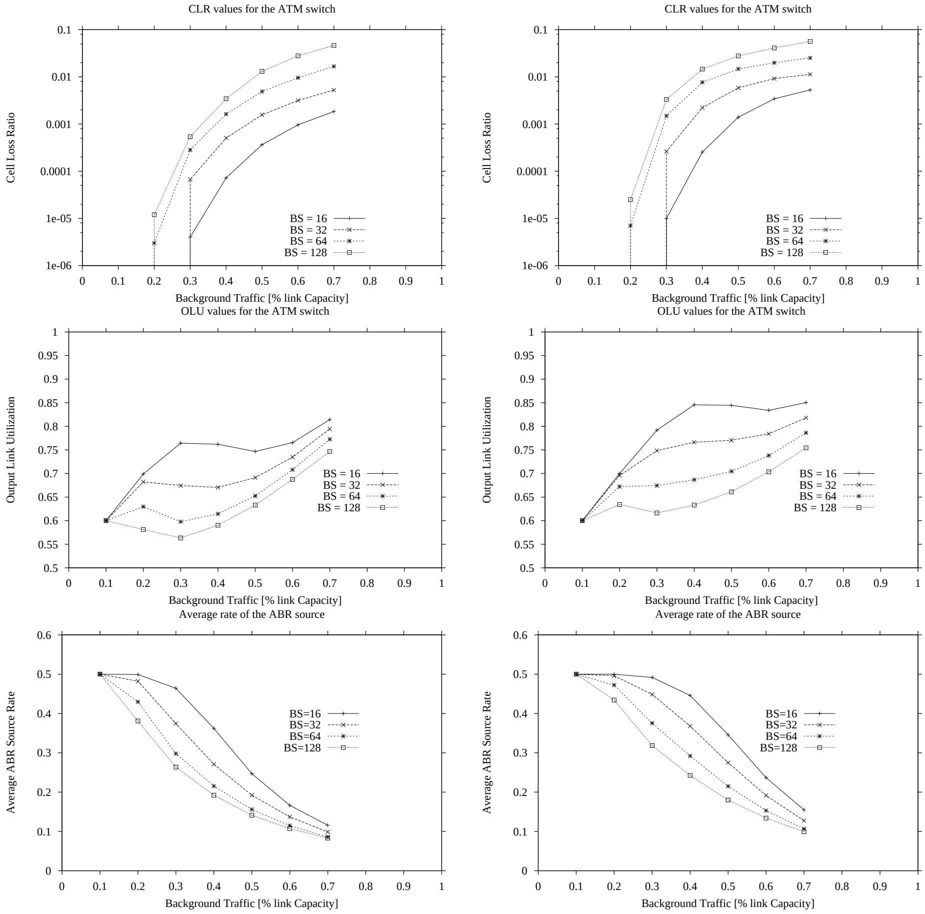

Figure 10: Cell loss ratio (top), output link utilization (middle), and average ABR source rate (bottom), as functions of the MMBP UBR traffic load, for either 2 ABR source rates (left) or 4 ABR source rates (right)

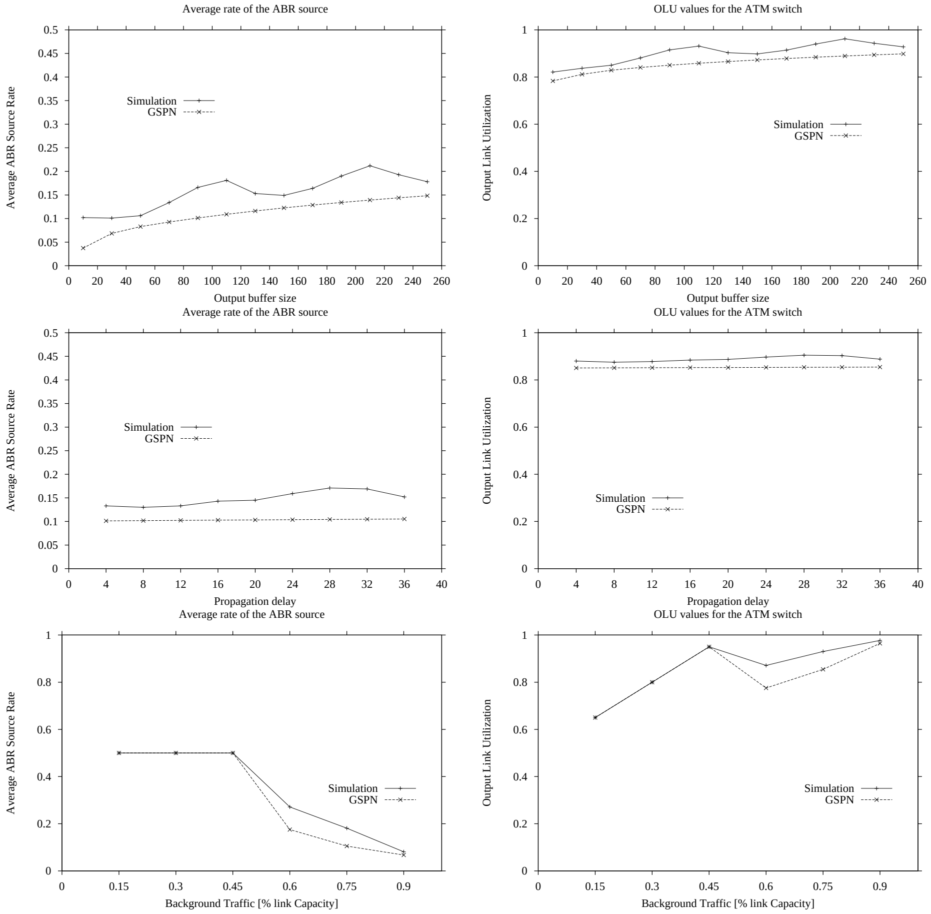

Figure 11: Increasing buffer size (top), increasing propagation delay (middle), and increasing Bernoulli UBR traffic load (bottom) – GSPN and simulation

with just two speeds, a careful study was conducted by comparing the GSPN results with those of a very detailed simulation setup. As already mentioned, simulation results were obtained with CLASS [12], a software tool for the simulation of ATM networks at the cell level that carefully implements all the details of the source, destination, and switch behaviors described in [11]. In order to force the ABR source to operate with just two speeds, the RRM parameters RIF and RDF were set to 1, the connection peak cell rate and initial cell rate (PCR and ICR) were both set equal to $C/2$, and the connection minimum cell rate (MCR) was set to $C/32$. The transfer of the feedback to the ABR source always uses in-rate backward RM cells.

As an example of the results of the validation of the GSPN models, Fig. 11 (top row) presents a comparison between the GSPN performance estimates and the results of a very detailed simulation of a 25 Mbit/s LAN where the distance between the ABR source and the congested switch is 54 km, and two UBR sources load with Bernoulli cell flows the same output channel used by the ABR source at 75% of its capacity (so that 25% of the LAN capacity on the average is available for the ABR source). The curves in the two graphs show the average ABR source rate (left column) and the utilization of the switch output link (right column) as functions of the output buffer size.

Moreover, the curves of the same performance indices are presented in Fig. 11 (middle row) when the buffer size is fixed to 100 cells and the distance between the ABR source and the switch is varied from 12 to 54 km, and in Fig. 11 (bottom row) when the buffer size is fixed to 100 cells, the distance between the ABR source and the switch is fixed to 54 km, and the UBR load of the output channel is varied from 0.15 to 0.9.

In all cases we can observe quite a good agreement between the GSPN performance predictions and the simulator estimates, and we can thus conclude that the GSPN model is accurate.

6 Conclusions and Future Work

An approach for the investigation of the performance of ATM networks by means of GSPN models was proposed, and an example of its application to the study of ATM LANs with users exploiting the UBR and ABR service categories was illustrated.

The adopted modeling approach is based on the use of just one timed transition that models the system clock, and numerous immediate transitions that describe the system operations.

Numerical results were obtained and in a number of cases validated against very detailed simulation experiments, proving the accuracy of the GSPN modeling approach.

The possibility of a modular approach in the development of the models of the ATM LAN may lead to approximate solution techniques where the GSPN modeling paradigm is exploited in conjunction with more traditional analytical approaches, thanks to the independence of the behaviors of the output subsystem and of the subsystem comprising the input interface and the switching fabric. Such hybrid approaches could lead to the possibility of a performance analysis of quite large configurations.

Acknowledgments

This work was supported in part by a research contract between Politecnico di Torino and CSELT, in part by the EC through the Copernicus project 1463 ATMIN, in part by the Esprit Human Capital and Mobility project MATCH, in part by the Italian Ministry for University and Research. K.Begain was supported by the Italian Government through the Italy-Mu'tah University cooperation project.

References

[1] M.Ajmone Marsan, G.Balbo, G.Conte, “A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems”, *ACM Transactions on Computer Systems*, Vol. 2, n. 2, May 1984, pp. 93-122.

[2] M.Ajmone Marsan, G.Balbo, G.Conte, S.Donatelli, G.Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, John Wiley & Sons, 1995.

[3] B.Haverkort, H.P.Idzenga, B.Kim, "Performance Evaluation of Threshold-Based ATM Cell Scheduling Policies Under Markov-Modulated Poisson Traffic Using Stochastic Petri Nets", *Performance Modelling and Evaluation of ATM Networks*, Editor: D.D.Kouvatsos, Chapman & Hall, pp. 551-572, 1995

[4] H.P.Idzenga, B.R.Haverkort, "Structural Decomposition and Serial Solution of Stochastic Petri Net Models of the Gauss Switch", *PNPM 95*, Durham, NC, USA, October 1995, poster paper.

[5] L.Kant, W.H.Sanders, "Loss Process Analysis of the Knockout Switch Using Stochastic Activity Networks", *ICCCN '95*, Las Vegas, NV, USA, September 1995.

[6] R.J.F.De Vries, "GAUSS: A Single-Stage ATM Switch With Output Buffering", *International Conference on Integrated Broadband Services and Networks*, 1990.

[7] Y.S.Yeh, M.G.Hluchyi, A.S.Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching", *JSAC*, Vol. 5, n. 8, 1987, pp. 1274-1283.

[8] J.F.Meyer, A.Movaghar, W.H.Sanders, "Stochastic Activity Networks: Structure, Behavior, and Application", *International Workshop on Timed Petri Nets*, Torino, Italy, July 1985.

[9] M.Ajmone Marsan, G.Chiola, "On Petri Nets with Deterministic and Exponentially Distributed Firing Times", *Advances in Petri Nets 1987*, LNCS n.266, Springer Verlag, 1987.

[10] G.Ciardo, C.Lindemann, "Analysis of Deterministic and Stochastic Petri Nets", *PNPM 93*, Toulouse, France, October 1993.

[11] ATM Forum/95-0013R8, "ATM Forum Traffic Management Specification", Version 4.0, April 1996.

[12] M.Ajmone Marsan, A.Bianco, T.V.Do, L.Jereb, R.Lo Cigno, M.Munafò, "ATM Simulation with CLASS", *Performance Evaluation,* Vol. 24, n. 1-2, November 1995, pp. 137-159.

[13] G.Chiola, G.Franceschinis, R.Gaeta, M.Ribaudo, "GreatSPN 1.7: GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets", *Performance Evaluation*, Vol. 24, n. 1,2, November 1995, pp. 47-68.