GSPN ANALYSIS OF ABR IN ATM LANS*

M.Ajmone Marsan[1], K.Begain[2], R.Gaeta[3], M.Telek[4]
[1] Dipartimento di Elettronica, Politecnico di Torino - Italy
[2] Department of Computer Science, Mu'tah University - Jordan
[3] Dipartimento di Informatica, Università di Torino - Italy
[4] Department of Telecommunications, Technical University of Budapest - Hungary
ajmone@polito.it, begain@nets.com.jo, rossano@di.unito.it, telek@hit.bme.hu

Abstract

This paper describes a GSPN model of an ATM LAN exploiting the EFCI ABR service category, and illustrates some numerical results. The GSPN model is constructed adopting a recently proposed approach for the use of GSPNs in the development of performance models of ATM network components. The numerical results quantify the dependence of the performance of the ATM LAN on several system parameters, such as the ATM switch buffer size, the feedback delay, and the threshold values used by EFCI to monitor network congestion. The analysis of the numerical results provides quite interesting insights that were not previously achieved with traditional simulative approaches.

1 Introduction

ATM, the Asynchronous Transfer Mode, is today regarded as one of the prominent technologies in both computer science and telecommunications. This stems from the fact that on the one hand the emerging broadband telecommunication networks for the integration of services, developed by the leading Telecom operators in many countries, and known with the acronym B-ISDN (Broadband Integrated Services Digital Network) are based on ATM, and on the other hand ATM has been adopted by many computer manufacturers as the new technology for high-speed LANs. The two factors combined have given enormous momentum to development of ATM products, as well as research into ATM systems, protocols, algorithms, and performance.

The performance analysis of ATM networks and their components at the cell level cannot in general be tackled with traditional continuous-time approaches, such as queuing networks or SPNs that were quite often adopted with success for the performance analysis of a wide variety of systems in a range of domains. The main reason for the inadequacy of the traditional performance analysis approaches is in the synchronous, discrete-time dynamics of ATM networks, when considered at the cell level. In spite of this fact, the GSPN modeling paradigm [1, 2, 3] can still be applied to the performance analysis of ATM networks at the cell level using the approach described in [4] and summarized below.

GSPNs comprise transitions whose firing delays can either be exponentially distributed random variables (exponential transitions) or deterministically equal to zero (immediate transitions). At a first glance, thus, the GSPN modeling approach appears to be totally unsuitable for the performance analysis of ATM networks at the cell level, mostly due to the constant cell transmission times. However, considering the fact that 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, and that also the time slots used inside ATM equipments are tied to the cell

*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.

rates on the transmission channels, we can conclude that in a model of the internal behavior of an ATM network or of one of its subsystems, delays between interesting events 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 clock is sufficient to drive the internal system dynamics. Furthermore, if we accept to also model in discrete time the user behavior, and thus the network workload, we can use the same clock that drives the internal system dynamics to also drive the dynamics of the workload model.

This would naturally lead to a PN model in which all transitions are immediate, with just one exception: one transition is timed with a constant delay defining the time unit in our model, which can thus be arbitrarily set to one. The stochastic process generated by the dynamic behavior of such a PN model is a discrete-time Markov Chain (DTMC), whose evolution over the state space is isomorphic to the marking process, and whose transition probabilities are computed from the reachable markings and from the weights of the enabled immediate transitions.

However, the fact that the only timed transition is associated with a constant or an exponentially distributed random delay makes no difference for the computation of a large quantity of interesting performance parameters, provided that the average firing delay of the exponential transition is set to one. This is due to the fact that while the PN model with the deterministic transition originates a DTMC, the PN model with the exponential transition originates a continuous-time MC; the relation between the two MCs is very tight: the discrete-time MC is the embedded MC of the continuous-time MC. The continuous-time MC is such that the average sojourn time in each state is equal to one, and thus its steady-state distribution is identical to the stationary distribution of its embedded MC [3]. We can thus compute the performance of ATM networks by developing appropriate GSPN models in which only one transition is timed with average firing delay equal to one, and all other transitions are immediate.

This GSPN approach to the analysis of ATM systems derives from a similar procedure that was first adopted in some unpublished works by G.Balbo for the investigation of multiprocessor systems, and whose application 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], as explained in [4].

This paper describes a GSPN model of an ATM LAN exploiting the ABR (Available Bit Rate) service category in its EFCI (Explicit Forward Congestion Indication) mode, and illustrates some numerical results, that quantify the dependence of the performance of the ATM LAN on several system parameters, such as the ATM switch buffer size, the feedback delay, and the threshold values used by EFCI to monitor network congestion.

A few previous examples of the use of PN-based approaches for the performance analysis of ATM networks exist in the literature [5, 6, 7, 8, 9, 10]. In [5], the authors develop a model of the Knockout switch using Stochastic Activity Networks (SANs), adopting an approach very similar to the one that we use for the representation of the deterministic cell transmission times. In [6] the authors use Deterministic and Stochastic PNs (DSPNs) [11, 12] for the analysis of Usage Parameter Control (UPC) at the ATM User-Network Interface (UNI). The paper [7] develops an SPN model to describe Available Bit Rate (ABR) traffic in ATM networks, and to determine the adequate size of buffers within ATM switches that handle ABR traffic. In [8] the authors use a GSPN approach to model and evaluate cell scheduling policies in ATM multiplexers. In [9] the authors develop a GSPN model of the Gauss switch approximating the deterministic cell transmission times with Erlang-10 distributions. In [10] a GSPN model of the Knockout switch is presented for the derivation of performance results for this type of ATM switch architecture.

This paper is organized as follows. Section 2 describes the system under investigation, and provides some details about the used ABR algorithm. Section 3 illustrates the GSPN models of the different parts of the system, as well as the complete GSPN model. Section 4 discusses numerical results, and finally some concluding remarks end the paper.

2 The ATM LAN

We consider an ATM LAN where sources are directly connected to a central ATM switch. This may correspond to a small LAN with end users directly connected to the central hub, or to the central part of a larger ATM LAN, where sources model groups of end users.

The central switch is taken to be non-blocking, so that the maximum number of cells that can arrive in the same slot at one output interface equals the number of input channels (say N).

The LAN users exploit either the UBR or the ABR transfer capability. The UBR (Unspecified Bit Rate) ATM transfer capability was designed for sources that cannot describe or control their cell transmission rates. The ABR (Available Bit Rate) ATM transfer capability was designed for sources that are able to adapt their cell rate to the network indications. Such sources are mainly personal computers or workstations, and are thus quite common in ATM LANs.

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 [13] for the control of ABR connections is based on the *Congestion Indication* (CI) and *No Increase* (NI) bits of 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 in turn is based on the occupancy of the output link buffer. Two thresholds are 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.

3 The GSPN Model

Since the heart of the ATM LAN is the central ATM switch, the system model focuses on this switch, considering users as cell sources with given characteristics.

The fact that the ATM switch is taken to be non-blocking allows considering one output channel at a time, since cells that arrive at an input port are brought to the output port interface with a constant delay, and no interference among cell flows destined to different outputs.

The GSPN model of the portion of the ATM LAN that refers to one of the switch output channels 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 model*)
- the cell generation processes due to ABR sources (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; they can be used to assemble a complete GSPN model for an ATM LAN with UBR and ABR sources. As an example, we shall show how a model can be constructed in the case of a small ATM switch with 4 input and output ports, 3 of which are loaded with UBR traffic, while the fourth is used by an 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 time unit within the ATM switch (which is taken to be equal to the slot time on input and output channels); we can think of *clock* as being a deterministic transition in the model

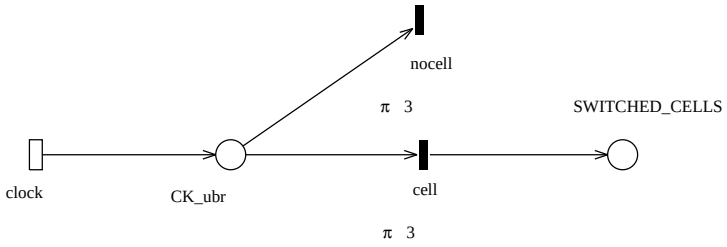

Figure 1: The UBR source model

description, in order to better understand the model, but it will be considered an exponential transition in the model solution, as we explained.

Transition *clock* always has concession, but it becomes enabled only when no (higher-priority) immediate transition is enabled. Thus, the evolution of the GSPN model alternately repeats two phases: the first phase comprises the enabling time of *clock* and the firing of this transition; the second phase consumes no time, and comprises the firing of several immediate transitions, terminating only when no more immediate transitions are enabled. When *clock* fires, one token is deposited in the places named CK_ABR, CK_UBR, CK_OUT, and CK_DELAY in the following subsections. The presence of tokens in these places activates the immediate transitions within GSPN model components. Throughout the paper we denote the priority j of an immediate transition t by drawing the symbol πj near t.

3.1 The Workload 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.

The description of the characteristics of sources exploiting the UBR service category is simply provided by a Bernoulli process, whereas the description of the behavior of the ABR sources must account for the feedback signals received from the switch.

3.1.1 The Bernoulli UBR source model

The GSPN model for one UBR source, corresponding to a Bernoulli generation of cells, is depicted in Fig. 1. The presence of a token in place CK_UBR enables the two conflicting immediate transitions *cell* and *nocell*. The firing of *nocell* (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 output interface, where it is deposited in an internal buffer. This is modeled by the token deposited in place SWITCHED_CELLS.

When more than one UBR 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 shown in Fig. 1. This obviously implies a reduction in the number of states of the GSPN model.

The aggregate GSPN model of K UBR sources comprises $K+1$ transitions 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

$$w_i = \left(\begin{array}{c} K \\ i \end{array} \right) p^i (1-p)^{K-i}$$

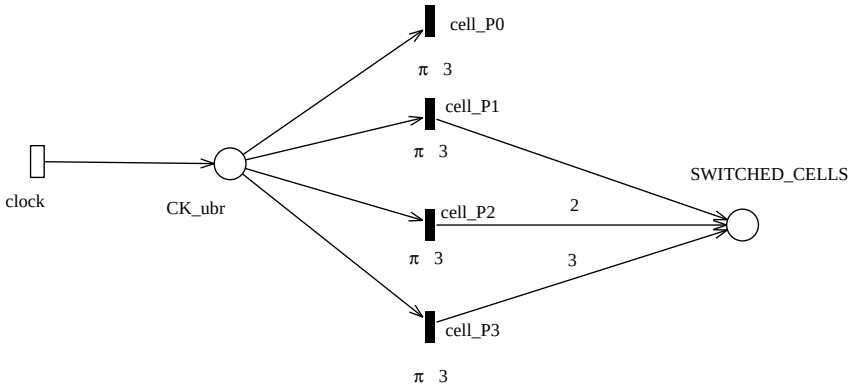

Figure 2: The aggregate UBR source model for three identical UBR sources

if all UBR sources have the same parameter p. Instead, when UBR 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 equal to 0.

As an example, Fig. 2 shows this aggregate model for three identical UBR sources.

3.1.2 The ABR source model

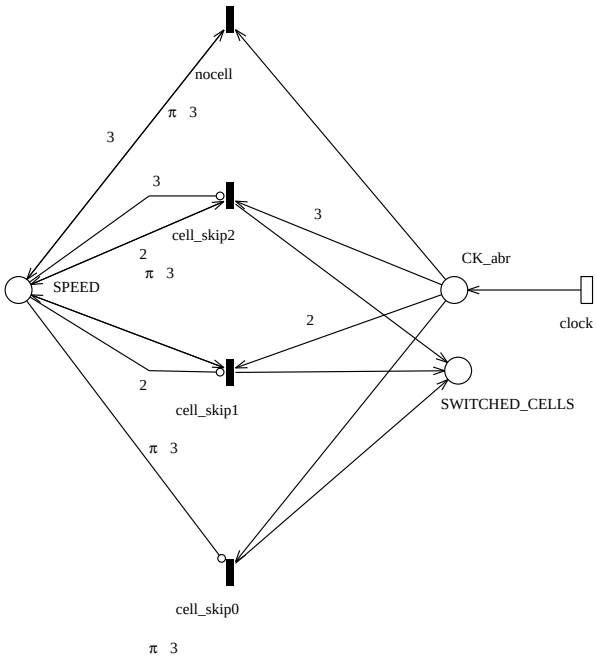

Figure 3: The ABR source model

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 systems, the possible cell rates of an ABR source are very numerous, but finite. In our 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 either equal to the link cell rate (this means that the ABR source transmits at full speed), or equal to zero (this means that the ABR source is silent), or equal to an integer divisor of the link rate (1/2, 1/3, 1/4, etc.). We denote the number of speeds greater than zero of an ABR source as ns.

Fig. 3 shows the ABR source model in the particular case in which the possible cell rates relative to the link speed are 1, 1/2, 1/3, and 0. 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_ABR, one of the four immediate transitions *cell_skip*0, *cell_skip*1, *cell_skip*2, *nocell*, may become enabled.

Transition *cell_skip*0 is enabled if place SPEED is empty, and place CK_ABR contains one token; its firing removes one token from place CK_ABR, and generates one token in place SWITCHED_CELLS. The marking of place SPEED is not altered. The firing of transition *cell_skip*0 thus models the generation of a cell at every clock time, hence the ABR source transmission at the link rate.

Transition *cell_skip*1 is enabled if place SPEED contains one token, and place CK_ABR contains two tokens; its firing removes two tokens from place CK_ABR, and generates one token in place SWITCHED_CELLS. The marking of place SPEED is not altered. The firing of transition *cell_skip*1 thus models the generation of a cell at every second clock time, hence the ABR source transmission at half the link rate.

Similarly, transition *cell_skip*2 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 quarter, one fifth, etc., of the link cell rate. However, this results in more complex behaviors, hence in increased reachability sets.

Finally, transition *nocell* is enabled if place SPEED contains three tokens, and place CK_ABR contains at least one token; its firing removes one token from place CK_ABR, and results in no cell generation. The marking of place SPEED is not altered. The firing of this transition thus models the silence state of the ABR source.

3.2 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.

The GSPN model of the output port queue is shown in Fig. 4.

The number of tokens in place OUTPUT_BUFFER_SPACE, initially set to the value M, indicates the available space in the output buffer, whereas the number of tokens in place OUTPUT_BUFFER indicates the number of cells awaiting transmission.

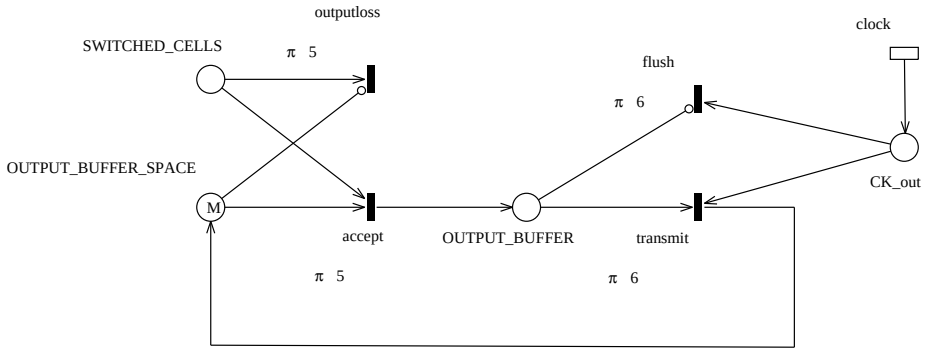

Figure 4: Output interface model

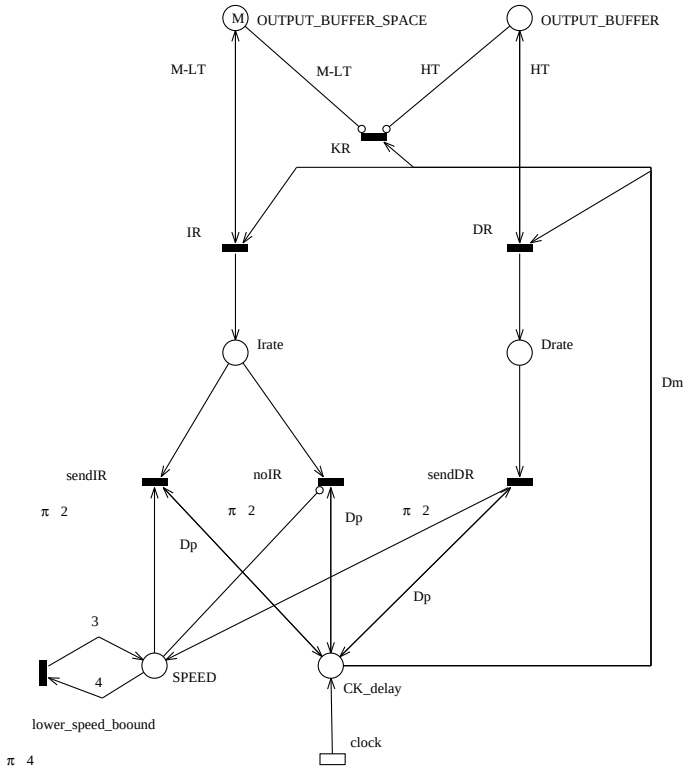

Figure 5: Model of Feedback Mechanics

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 *outputloss* or *accept* can be enabled. The firing of *outputloss* 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.

3.3 The feedback model

According to the EFCI ABR scheme, an ATM switch determines its congestion status depending on the buffer occupancy, and issues the feedback to the ABR source accordingly.

As mentioned before, in our model the number of tokens in place SPEED determines the cell rate of an 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. 5 shows the GSPN model for the EFCI 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.

The three mutually exclusive immediate transition IR, KR, and DR in Fig. 5 have priority 1, and 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.

Transition DR corresponds to a high congestion situation, and is enabled when the buffer occupancy exceeds the high threshold HT; this is modeled by the test arc with multiplicity HT from place OUTPUT_BUFFER to transition DR. Firing this transition does not alter the marking of OUTPUT_BUFFER, but generates one token into place $Drate$ and removes the Dm tokens from place CK_DELAY.

Transition IR corresponds to an underutilization situation, and is enabled when the buffer occupancy is below the low threshold LT; in this case the multiplicity of the test arc from place OUTPUT_BUFFER_SPACE to transition IR is $M - LT$, where M is the buffer size, and LT is the low threshold. Firing this transition does not alter the marking of OUTPUT_BUFFER_SPACE, but generates one token into place $Irate$ 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 a Keep Rate 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 requires 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.

When this happens, two possibilities arise, depending on the result of the first phase: 1) one of the two transitions *sendDR* and *sendIR* (both are connected with a test arc of weight Dp to place CK_DELAY) is enabled, or 2) both are not enabled.

If the first phase resulted in the generation of one token in place $Drate$, then transition *sendDR* will remove that token and add new token to place SPEED in order to decrement the cell rate of the ABR source, as described in Section 3.1.2. Should the addition of the new token to place SPEED result in a speed smaller than the permitted values, transition *lower_speed_bound* becomes enabled and fires removing one token from place SPEED, thus preventing the number of tokens to increase above the allowed value.

If instead the first phase ended with one token in place $Irate$, then transition *sendIR* will be enabled and fire only if at least one token exist in place SPEED; the firing of *sendIR* removes one token from place SPEED as well as the token from place $Irate$, thus producing an increase in the transmission rate of the ABR source only if this is possible. Otherwise, i.e. if the ABR source is already transmitting at full speed (the number of tokens in place SPEED is zero), then transition *noIR* will fire, removing the token from place $Irate$.

Obviously, transitions *sendDR*, *sendIR*, and *noIR* must have higher priority than transitions DR, IR, and KR so that the second phase of the previous control cycle can be finished before starting the next cycle. Thus, their priority is set to 2.

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 of the model allows considering longer propagation delays (at the expense of an increase in the reachability set cardinality).

3.4 The Full GSPN Model

The full 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.

As an example, the full GSPN model of a small ATM LAN comprising a central switch with 4 input and output ports, 3 of which are loaded with UBR traffic, while the fourth is used by an ABR source is shown in Fig. 6.

It can be immediately observed that this model results from the composition of the submodels

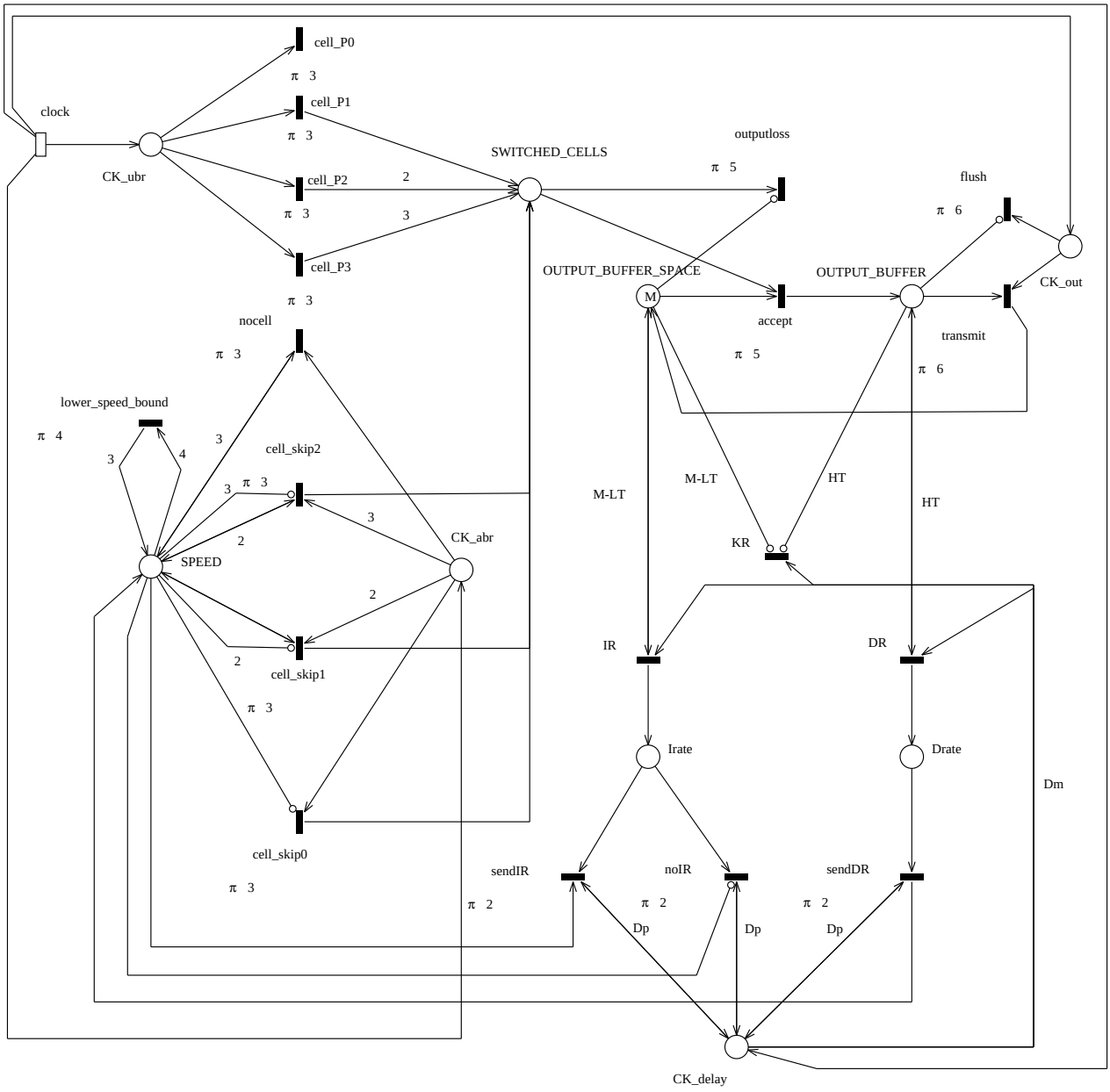

Figure 6: The full GSPN model of the ATM LAN

that were previously described; we therefore omit its description.

It is worthwhile observing that 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. 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 and finally, the last set of operations concerns the ABR feedback mechanism model, which comprises transitions at the lowest priority levels.

4 Numerical Results

We will now present and discuss numerical results obtained from the GSPN models of the previous section using the GreatSPN software tool [14] on a Sun SPARC 10 workstation equipped with 80 MBytes of main memory and running the SunOS 4.1.3 operating system.

We focus on four performance indexes:

- *Cell Loss Probability*, defined as
$$\frac{X(outputloss)}{\rho}$$
where $X(t)$ is the throughput of immediate transition t, and ρ is the offered load to the output link:
$$\rho = \sum_{i=1}^{3} i \cdot X(cell\_P_i) + \sum_{j=0}^{ns-1} X(cell\_skip_j)$$

- *Output Link Utilization*, defined as $X(transmit)$

- *Mean Number of Queueing Cells* defined as the average number of tokens in place OUTPUT_BUFFER

- *ABR Source Mean Percentage Speed* defined as
$$\sum_{i=0}^{ns-1} \frac{100}{i+1} \cdot p_i$$
where
$$p_i = \frac{X(cell\_skip_i)}{X(nocell) + \sum_{j=0}^{ns-1} X(cell\_skip_j)}$$

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

We consider a small ATM LAN based on a central switch with 4 input and output ports, and focus our interest on one particular output port, that is assumed to receive from 3 of the input ports UBR traffic amounting to 60% of the capacity of the output link, and from the fourth input port the ABR traffic generated by one ABR source.

The first set of results is presented in the four graphs of Fig. 7, that show the cell loss probability (upper left graph), the output link utilization (upper right graph), the average number of cells queued in the output buffer (lower left graph), and the average percentage speed of the ABR source (lower right graph), as functions of the propagation delay from the ABR source to the ATM switch, measured in cells. The ATM switch output buffer size is taken to be 100 cells, and the interval between two ABR feedback generations is taken to be 36 cells (the value suggested in [13] is 32). The number of speeds of the ABR source is varied between two (the ABR source either transmits at full speed or is silent) and five (the ABR source speeds relative to the output link capacity are 1, 1/2, 1/3, 1/4, and 0). The low and high thresholds of

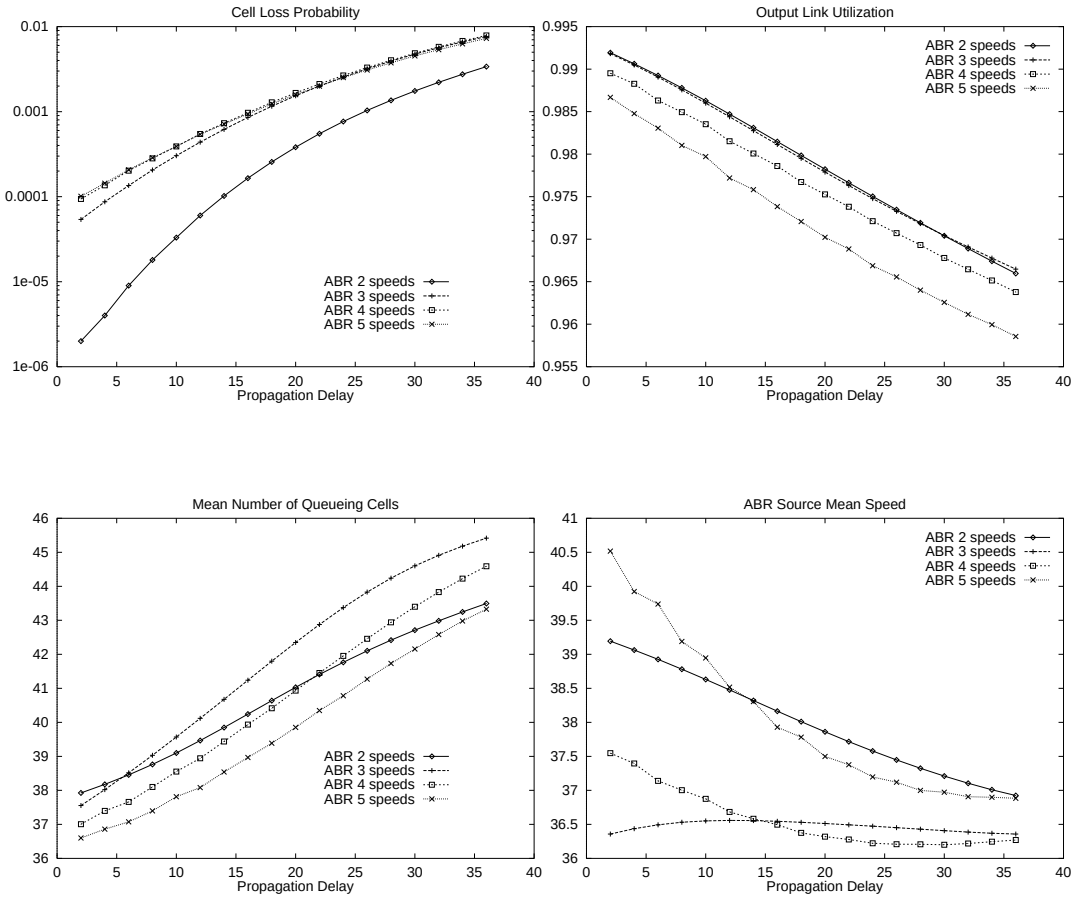

Figure 7: Cell loss probability (upper left), output link utilization (upper right), average number of cells queued in the output buffer (lower left), and average percentage speed of the ABR source (lower right), versus the propagation delay from the ABR source to the ATM switch, for different numbers of speeds of the ABR source; ATM switch output buffer size 100 cells, interval between two ABR feedback generations 36 cells, low threshold 10, high threshold 60.

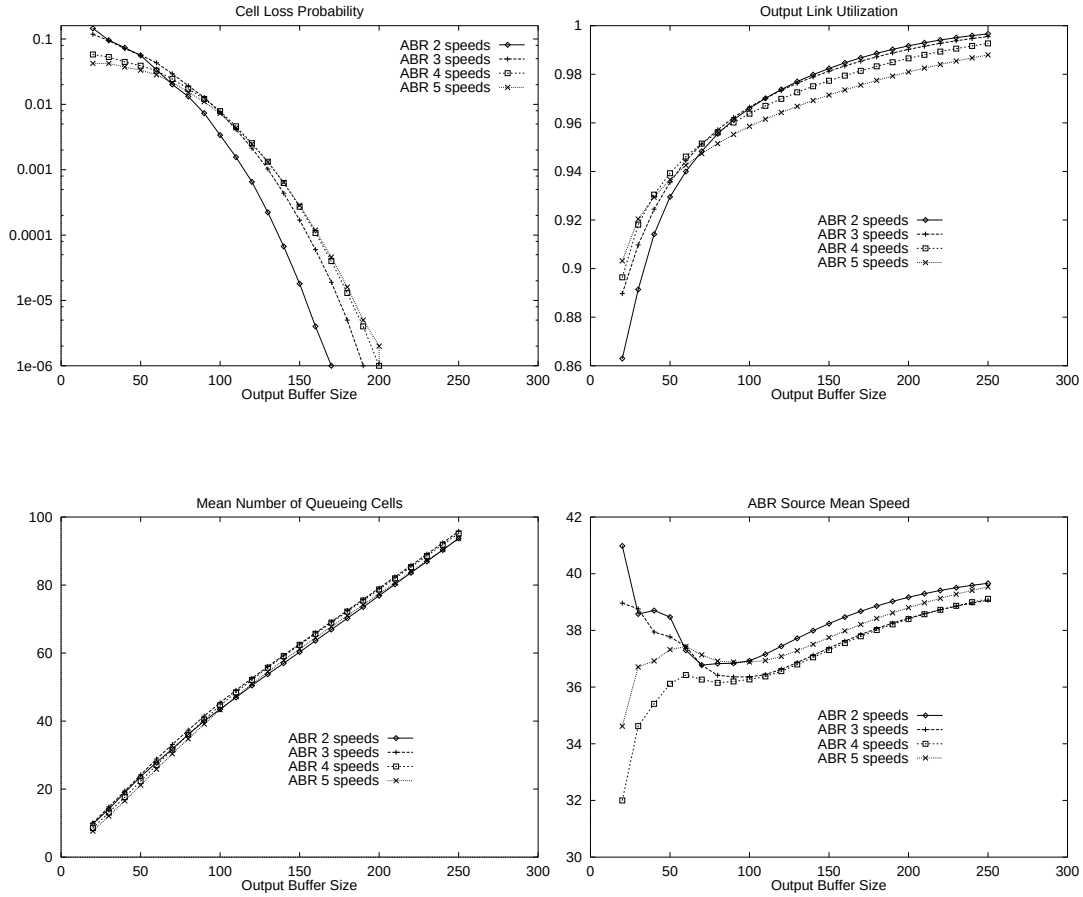

Figure 8: Same as Fig.7, but for $D_p = 36$, and variable output buffer size

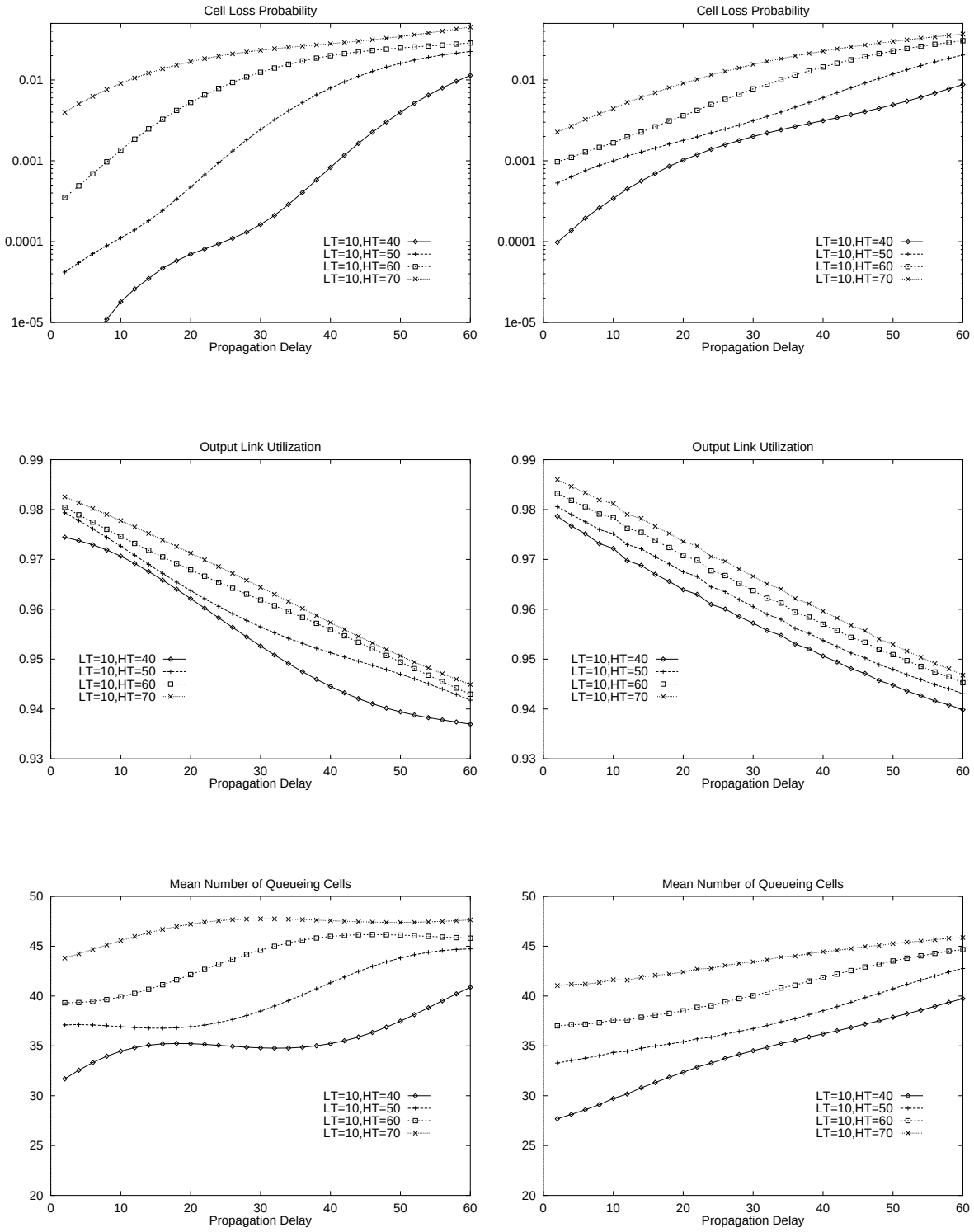

Figure 9: Cell loss probability (top row), output link utilization (middle row), and average number of cells queued in the output buffer (bottom row), for either 2 (left column) or 5 (right column) speeds of the ABR source, versus the propagation delay from the ABR source to the ATM switch, for different values of the high threshold, with low threshold 10; ATM switch output buffer size 100 cells, interval between two ABR feedback generations 60 cells.

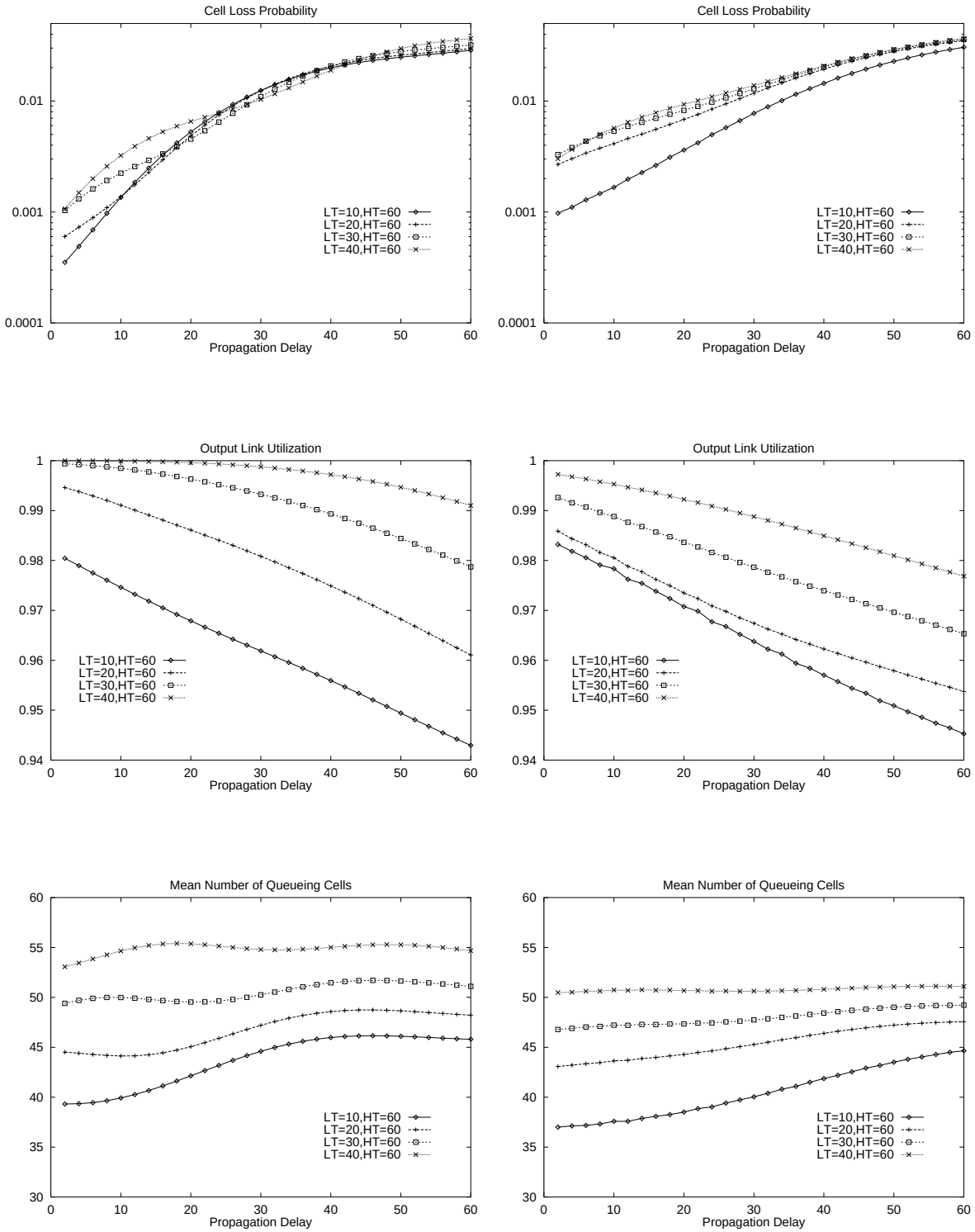

Figure 10: Same as Fig. 9, but high threshold 60 and variable low threshold

the output buffer are set to 10% and 60% of the buffer size, respectively (which in this particular case means 10 and 60 positions). The propagation delay from the ABR source to the ATM switch is varied from 2 to 36 cells. Translating these values into distances depends on the considered link data 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 the cell transmission time is about 17 μs, and thus corresponds to about 3 km. In the former case the propagation delay values (from 2 to 36 cells) correspond approximately to distances between 1 and 18 km; in the latter case to distances between 6 and 108 km.

From the curves we observe quite an interesting effect, that was not previously discussed in the literature: when the ABR source is bounded to an ON-OFF behavior by the existence of only two possible speeds, the system performance is significantly better than with a larger number of speeds. Indeed, lower cell loss probabilities and higher utilizations of the output link are simultaneously achieved, with comparable buffer occupancy, and with an average percentage speed that is similar to the one obtained with 5 speeds, but significantly larger than those obtained with either 3 or 4 speeds.

In order to better investigate this phenomenon, we studied how the same performance parameters behave as functions of the output buffer size. Results are plotted in Fig. 8, fixing the propagation delay to be equal to 36 cell times (18 km for a 155 Mbit/s LAN — 108 km at 25 Mbit/s). It is rather evident that differences in performance can be significant in the case of small buffers, but tend to disappear for increasing buffer size. The configuration with only 2 speeds for the ABR source behaves poorly for very small buffers (note the high cell loss probabilities and low utilizations), but tends to the best performance for increasing buffer size. It should be observed that in realistic ATM LANs the buffer sizes can be quite large, so that the configuration with just 2 ABR speeds could be advantageous in terms of performance (it surely has the big advantage of being very simple to implement, much simpler than the currently proposed approaches with a very large number of possible speeds). This is quite an interesting conclusion, that obviously will require much deeper investigations in a number of different networking scenarios, but shows the usefulness of the GSPN approach to the study of ATM systems: adopting an analytical (even if requiring numerical solution) rather than a simulative approach reduces the cost for the computation of the performance indices, and thus allows more freedom for the investigation of the influence of system parameters on performance.

Another set of parameters whose influence on system performance is difficult to investigate through simulation are the positions of the two thresholds in the output buffer.

In Fig. 9 we show the cell loss probability (top row), the output link utilization (middle row), and the average number of cells queued in the output buffer (bottom row), for either 2 (left column) or 5 (right column) speeds of the ABR source, versus the propagation delay from the ABR source to the ATM switch, for different values of the high threshold, with low threshold 10, for ATM switch output buffer size 100 cells, and interval between two ABR feedback generations 60 cells.

In Fig. 10 we show curves for the same parameters in the same configurations, keeping the high threshold equal to 60, and varying the low threshold.

In this case the numerical results indicate that setting the thresholds to low values has a beneficial effect on cell loss probabilities and on buffer occupancy (this is rather obvious, since lowering the thresholds makes the ABR algorithm more conservative), but implies a reduction in the utilization of the output channel (obvious again, because of the reduction of the ABR source speed). High utilizations are achieved with large thresholds, at the cost of more cell losses.

It is also worthwhile noting that the cell loss probability is more sensitive to changes in the high threshold, while the link utilization is more sensitive to the low threshold. This is due to the fact that cell losses occur when the buffer fills because the ABR mechanism is not fast enough to slow down the source when congestion builds up. If the high threshold is lower, the congestion signal is generated sooner, and more time is available to slow down the ABR source. Instead, the link underutilization derives from the output buffer remaining empty because the

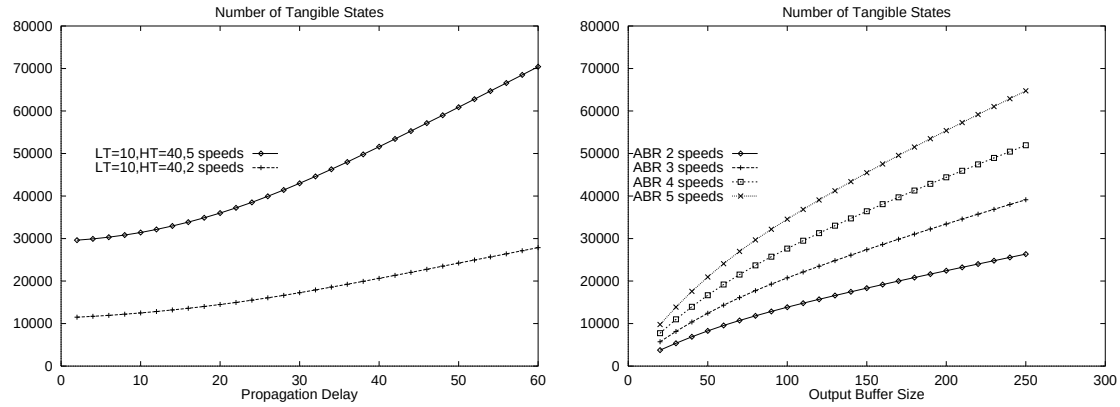

Figure 11: Number of tangible states for the graphs of Fig. 9 (left) and of Fig. 8 (right).

ABR mechanism is not fast enough to allow the source to increase its rate when the buffer is emptying.

From these insights it could be concluded that the two thresholds should be set close to one another, or even substituted with just one threshold in the proximity of half the buffer size. Also in this case, our conclusion is rather innovative, since the choice of using two thresholds rather than just one, and of spacing them far apart, is generally due to the desire to reduce the oscillations in the source speed with the introduction of a large hysteresis between the two thresholds, ultimately aiming at better performance. However, at least in our very simple setup, this approach appears not to be the most effective. Also in this case, the investigation of more complex scenarios will be necessary to confirm these indications.

As a final remark, it is important to observe that the computational cost for the derivation of numerical results is quite small: for instance, the CPU time needed to obtain all the results plotted in Fig. 9 for $LT = 10$, $HT = 40$ and two speeds was 36 minutes, while the CPU time for the same model configuration and five speeds was 95 minutes. As another example, the CPU time needed to obtain the curves in Fig. 8 ranged from 33 minutes for the ABR source with two speeds to 90 minutes for the same system with five speeds. Please note that the machine we used for these experiments was shared among at least six users during the computations.

We also plotted in Fig. 11 the number of tangible markings generated from the models representing these systems. It can be easily noted that the number of tangible states increases as the number of ABR speeds increases. More interestingly, the system with an ON-OFF behavior, that in this preliminary investigation appears to be the most effective, has a rather small number of states that will allow us to consider more complex networking scenarios, e.g., more UBR Bernoulli sources, Markov Modulated UBR sources, larger output buffer sizes, longer propagation delay.

5 Conclusions and Future Works

This paper presented a GSPN model for the performance analysis of ATM LANs exploiting the UBR and ABR service categories for the provision of connectionless or best effort communication services. 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.

The numerical solution of the GSPN model of a simple ATM LAN allowed the observation of some interesting and rather unexpected behaviors that deserve a further investigation.

Most notably, in the considered setup, an EFCI ABR control technique that either allows the ABR source to transmit at full speed, or forces it to remain silent, proved to perform best.

This finding needs to be confirmed with the investigation of more complex setups, comprising a larger number of ABR sources, and possibly more complex network configurations. This further study will be conducted with the proposed GSPN approach as long as the resulting models will generate reachability sets of manageable size, so that a numerical solution of the corresponding Markovian models is possible. For larger reachability sets it will be necessary to resort to simulation.

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] G.Chiola, M.Ajmone Marsan, G.Balbo, G.Conte, "Generalized Stochastic Petri Nets: a Definition at the Net Level and its Implications," *IEEE Transactions on Software Engineering*, Vol. 19, n. 2, February 1993, pp. 89-107.

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

[4] M.Ajmone Marsan, R.Gaeta, "GSPN Models for ATM Switches", Submitted for publication

[5] L.Kant, W.H.Sanders, "Loss Process Analysis of the Knockout Switch Using Stochastic Activity Networks", *ICCCN '95*, pp. 344-349, 1995.

[6] B.Müller-Clostermann, "DSPN-Modelling of Usage Parameter Control in ATM Networks", *Performance and Dependability Modeling with Stochastic Petri Nets*, Dagstuhl, 1995.

[7] A.Puliafito, M.Balakrishnan, K.S.Trivedi, I.Viniotis, "Buffer Sizing of ABR Traffic in an ATM Switch", *ICC '95*, Seattle, USA, pp. 316-320.

[8] 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

[9] 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.

[10] M.Ajmone Marsan, R.Gaeta, "GSPN Models of the Knockout ATM Switch", *4th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley,UK, 1996.

[11] M.Ajmone Marsan, G.Chiola, "On Petri Nets with Deterministic and Exponentially Distributed Firing Times", in: G.Rozenberg (editor), *Advances in Petri Nets 1987*, Lecture Notes in Computer Science n.266, Springer Verlag, 1987.

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

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

[14] G.Chiola, G.Franceschinis, R.Gaeta and 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.