Approximate Models for the Output Process of an ATM Multiplexer with Markov Modulated Input

Nelson L.S. Fonseca and John A. Silvester

Relatório Técnico IC-97-02

Janeiro de 1997

Approximate Models for the Output Process of an ATM Multiplexer with Markov Modulated Input

Nelson L. S. Fonseca
State University of Campinas
Institute of Computing
P.O. Box 6176
13083-970 Campinas SP
Brazil
e-mail: nfonseca@dcc.unicamp.br
Phone/fax: +55+19+2530123

John A. Silvester
University of Southern California
Electrical Engineering - Systems Department
Los Angeles, CA 90089-2562
U.S.A.

Abstract

The traffic in the future Broadband Integrated Digital Networks will be highly correlated and neglecting its correlations leads to a dramatic underestimation of its performance. In order to completly specify a queueing network framework, we need to define the stochastic processes resulting from the departure of a queue splitting and merging. In this paper we introduce a procedure for modeling the output process of a multiplexer with Markov modulated input and extend this procedure to model ATM multiplexers with selective discard mechanism. Moreover,we show frameworks for queueing networks with Markov modulated flows which can be used to estimate end-to-end performance in ATM networks.

I) Introduction

Queueing network models are of paramount importance for dimensioning communication networks [1]-[2]. Depending on the time scale of interest, different queueing network frameworks can be used to analyze multimedia networks. For instance, if one is interested in evaluating the network at the call level, product form networks can be used [3]-[4]. However, to analyze traffic control mechanisms at the burst/cell level a detailed description of the network flows is needed. To completely specify a queueing network framework we need to define the stochastic process resulting from: i) departures from a queue (output process), ii) splitting of a process due to routing and iii) merging of processes which go to the same queue (joining). The traffic in the future Broadband Integrated Services Digital Network will be highly correlated and neglecting these correlations leads to a dramatic underestimation of the delay and loss rate. The departure (output) process of an ATM multiplexer is also correlated [5]-[6] and an accurate representation of the output process is the first step towards the definition of queueing networks for ATM networks. In this paper, we introduce an approximate model for the output process of an ATM multiplexer and extend our analysis to include selective discard mechanism. Moreover, we describe frameworks for queueing networks with Markov modulated flows.

Recently, there has been a great deal of interest in using long-range dependent processes to model networks traffic. However, the impact of long-range dependent processes on network dimensioning is yet to be fully determined. Depending on the parameters, accurate performance results can sometimes be obtained using short to medium range processes (such as Markov modulated processes) [7]. Ongoing research is investigating the limits of applicability of Markov modulated processes approximations to long range dependent processes (and their queueing behavior) over network dimensioning [8].

We assume that the input traffic to the queueing network is modelled as a Discrete Time Batch Markovian Arrival Process (D-BMAP) [9]-[11]. An ATM multiplexer is viewed as a finite buffer queue with FCFS services (D-BMAP/D/1/k). We model the output pro-

cess of a multiplexer as a two-state Markov Modulated Bernoulli Process (MMBP) [12]. Insofar as the MMBP is a sub-case of the more general D-BMAP, we are able to maintain an uniform representation of the flows in a queueing network. We also extend this two-state MMBP model to represent the output process of a queue with prioritized input process (D-BMAP$^{[H,L]}$/D/1/K) [13] and define a framework for queueing networks with prioritized flows.

This paper is organized as follow. In section II, we briefly describe previous work. Section III shows a framework for queueing networks with Markov modulated flows. Section IV introduces an approximate model for the output process of a D-BMAP/D/1/K queue. Section V and VI respectively describe a framework for queueing networks with prioritized flows and a procedure for the analysis of the output process of a multiplexer with selective discard mechanism. Section VII presents some network examples and finally some conclusions are drawn in section VIII.

II) Previous Work

We can roughly categorize the work done so far into two groups. The first group models the output process of a queue fed by several on/off sources as an on/off process (a renewal process) and sets its parameters so as to approximate the correlation structure. Ren et al. [14] defined the *on/off* process with the *on* period duration that take into consideration the difference of the unfinished work between the transition *on* to *off* and *off* to *on* epochs of the input process. Frost and Wang [15] determined the *on/off* source parameters by matching the “age distribution” (duration of *on* periods) of the *on/off* source with the age distribution (duration of busy periods) of the output process. Lau and Li [16] evaluated by simulation the distortion effect of an individual *on/off* sources when it visits a statistical multiplexer. They found out that when the peak rate of an *on/off* source is less than five percent of the link capacity the distortion is negligible.

The second group is the work that study the output process of queues with Markov Modulated input [6], [12], [17]-[18]. Saito [6] studied the output process of the N/G/1 queue and particularly of the MMPP/D/1 queue. By comparing the z-transform curves of

the covariance of interarrival times for both input and output processes of a queue, Saito concluded that covariances are likely to be preserved. Takine et al. [17] derived expressions for the k^{th} moment of the interdeparture time and the statistics of busy and idle periods of a D-BMAP/D/1/K queue. Park et al. [18] proposed a procedure for matching the output process of a 2-MMBP/Geo/1/K queue with the statistics of a two state Markov Modulated Bernoulli Process (2-MMBP). Our work [12] differs from Takine et al. [17] in the sense that they derive expressions for the k^{th} moment of the interdeparture time while ours gives an approximate representation of the output process. Our approximate model was derived simultaneously with Park et. al's work and the results are quite similar. However, none of this previous work has considered networks with prioritized flows.

III) Queueing Networks with Markov Modulated Flows

We assume that the input traffic of the queueing network is modelled as a Discrete Time Batch Markovian Arrival Process (D-BMAP). The discrete time assumption derives from the ATM standard. In a Discrete Time Batch Markovian Arrival Process (D-BMAP) at each discrete time a batch may arrive. The batch size distribution is a function of an underlying Markov chain. A D-BMAP is completely specified by the matrices D_n whose elements $(d_{ij})_n$ give the probability of going from state i to state j and having a batch arrival of size n [9].

In our investigation, we consider open queueing networks. At each node there is a single server with finite buffer space and constant service time. Service is provided in a First-Come-First-Served fashion. In order to solve this queueing network with non-renewal flows, we employ the parametric decomposition approximation which is a generalization of the product form type of solution [19]-[20]. Lau and Li [16] have recently validated the nodal decomposition (parametric decomposition) for networks with integrated traffic.

To complete specify a queueing network framework we need to define the stochastic process resulting from: i) departures of a queue (output process), ii) splitting of a process

due to routing and iii) merging of processes which go to the same queue (joining). We define the network flow operators as:

Output process

At each slot, there is at most one departure from the queue, and departure slots are correlated. In addition to that, the output process of a D-BMAP/D/1/K queue is correlated. Thus, we represent the output process as a Markov Modulated Bernoulli Process. We develop a procedure for matching the statistics of the output process with the statistics of a two-state MMBP (we focus our attention on a two-state representation due to its low computational complexity - section IV). Moreover, by modelling the output process as a MMBP, we are able to represent all the flows in the network as D-BMAP processes.

Joining

The superposition of two D-BMAP processes with m_1, m_2 states and n_1, n_2 maximum batch size is also a D-BMAP with $m_1 \times m_2$ states and $n_1 + n_2$ maximum batch size. The matrix D_k whose elements $(d_{ij})_k$ which give the probability of going from state i to state j and having a batch arrival of size k are computed as:

$$D_k = \sum_{q=0}^{min\ (k, n_1)} D_q^{(1)} \otimes D_{k-q}^{(2)}$$

where $A \otimes B$ denotes the Krockener product of matrix A by matrix B.

Splitting

We assume that routing is state independent which means that the probability of a cell departing from one node and going to another node is fixed. When characterizing the flow between two nodes, we represent the output process of the first queue as an MMBP process with parameters (p_1, p_2, α_1, α_2); and then we model the flow that goes to the second queue as an MMBP with parameters ($p_1 \times p_{ij}$, $p_2 \times p_{ij}$, α_1, α_2); where:

p_n (n=1,2) is the probability of having an arrival in state n

α_n is the transition probability in state n

p_{ij} is the probability that a cell leaves node i and goes to node j.

Alternate splitting operators can be developed to capture the behavior beyond the memoryless splitting as described in [21].

IV) An Approximate Model for the Output Process of a D-BMAP/D/1/K Queue

The output process of a queue with Markov modulated inputs is a correlated single arrival process. We are able to exactly represent the output process as a Markov Modulated Bernoulli Process if we define the underlying Markov chain state as being the number of cells in the system plus the state of the input process. For instance, If we have a gated server (i.e., if a cell finds the server empty at its arrival slot, it can only be transmitted at the next slot) then, the matrices D'_0 and D'_1 are given by [9]:

$$D'_0 = \begin{bmatrix} D_0 & D_1 & \dots & D_{k-1} & \sum_{n=k}^{\infty} D_n \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$D'_1 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ D_0 & D_1 & D_2 & \dots & D_{k-1} & \sum_{n=k}^{\infty} D_n \\ 0 & D_0 & D_1 & \dots & D_{k-2} & \sum_{n=k-1}^{\infty} D_n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & D_0 & \sum_{n=1}^{\infty} D_n \end{bmatrix}$$

Unfortunately, to represent the output process as an exact MMBP is computationally unfeasible since the number of states grows as a function of the buffer size and the number of states of the input process. Thus, we represent the output process with two states and match the following statistics of the exact process with the statistics of our reduced model:

i) Mean

ii) Variance

iii) Covariance at lag = 1

iv) Covariance at lag = 2

For a D-BMAP, the mean, the variance and the covariance at lag *k* can be computed as:

$$\lambda = \pi\left(\sum_{k=1}^{\infty} kD_k\right)\bar{e}$$

$$var = \pi\left(\sum_{k=1}^{\infty} k^2 D_k\right)\bar{e} - \lambda^2$$

$$cov(x_1, x_k) = \pi\left(\sum_{n=1}^{\infty} nD_n\right)D^{k-2}\left(\sum_{n=1}^{\infty} nD_n\right)\bar{e} - \lambda^2$$

where $\bar{e}$ is the unit column vector and π is the steady state probability of the underlying Markov chain, i.e;

$$\pi D = \pi \qquad \pi\bar{e} = 1$$

To validate the matching procedure, we consider two queues in tandem. The input to the first queue is a two-state D-BMAP as defined in [17]. The input to the second queue is composed of the output process from the first queue and an interfering process. This interfering process is introduced in order to avoid the “non-queueing” phenomenon in tandem queues with constant service time. To assess the accuracy of the matching procedure, we compare the mean delay and the loss probability at the second queue when the output process is substituted by a two-state MMBP with the results produced by a simulation experiment (Figure1). In the simulation experiments, we use the independent replication method to find a 95% confidence interval. We report the percentage error defined as ($|X_{sim} - X_{match}| / X_{sim}$) x 100 where X_{sim} and X_{match} are respectively the results produced by the simulation experiment and by the matching procedure.

The input to the first queue and the interfering process are two-state D-BMAP with the same transition probability in each state (α). The batch size is Poisson distributed with mean $(1+c)\rho$ (state 1) and $(1-c)\rho$ (state 2) where ρ is the overall traffic intensity and c is a parameter. It was demonstrated in [22] that the square coefficient of variation (C_v^2) and the correlation coefficient of the number of arrivals at lag n ($C_c(n)$) are respectively given by:

$$C_v^2 = \rho^{-1} + c^2$$

$$C_c(n) = \frac{c^2\rho}{1 + c^2\rho} \times (2\alpha - 1)^n$$

The data shown is this section corresponds to a server with gated service and buffer size 100. Time is normalized to one slot which has the same duration of a service time. In order to validate the accuracy of the computational procedure over a wide range of delay values, we vary the input parameters in a way such that we obtain the desired value at the second queue. Table 1 presents some results from our experiment. Overall, the percentage error of the delay estimation are under 7% [12].

To evaluate the impact of the input process mean arrival rate, variance and correlation, we keep constant two of three input parameters: ρ, α and c, and vary the third one. The parameters of the interfering process are set in a way to avoid the “non-queueing” phenomenon in tandem networks with constant service time. Regarding the delay estimation, Figure 2 shows the accuracy of the matching procedure increases as the offered load increases. We note that a 5% difference when ρ varies from 0.4 to 0.9. When varying c, we observe that for positively correlated streams, the procedure provides slightly more precise results for higher values of the coefficient of variance than it does for lower ones. Differences are under 2%. This trend did not emerge for negatively correlated streams. When varying α, we also observed that the correlation coefficient does not affect the accuracy of the procedure besides the impact just mentioned.

Regarding the loss rate estimation, Table 2 shows that the matching procedure is more precise for the estimation of higher values of the loss rate than it is for lower values. Differences are under 6% (Figure 3 also illustrates this trend). Figure 4 indicates that the precision increases with the coefficient of variance. We note that the maximum difference is under 5%. No significant impact on the accuracy of the procedure was observed as a function of the correlation coefficient.

For fixed values of the input process parameter we vary the interfering process parameters (ρ, c, α) and we noticed that the interfering process parameters did not impact the accuracy of the delay and loss rate estimation. We also investigated the procedure precision as a function of the buffer size (from 50 to 200). No significant impact on the accuracy was observed.

More Extensive validation data can be found in [22].

V) Queueing Networks with Prioritized Markov Modulated Flows

We now extend our queueing network framework in order to model communication networks with a selective discard mechanism at the cell level [23]-[29]. We consider that at each node, the buffer space is organized in a complete sharing fashion with push-out. We define a prioritized Discrete Time Batch Markovian Arrival Process (D-BMAP $^{[H, L]}$) as a D-BMAP process where the arrivals can be classified as either high or low priority. In a D-BMAP $^{[H, L]}$ the probability that an arrival (cell) belongs to a certain priority class (priority probability) is independent of other cells and is a function of the state of the underlying Markov chain. A D-BMAP $^{[H, L]}$ is completely specified by the matrices D_n and by the vector $\vec{p}_{high}$ whose i^{th} component gives the probability of a cell being high priority when the underlying Markov chain is in state i. $p_{high} = \vec{p}_{high} \cdot \vec{\pi}$ gives the unconditional high priority probability. The i^{th} component of $\vec{p}_{low}$ is 1 - the i^{th} component of $\vec{p}_{high}$. The prioritized Markov Modulated Bernoulli Process (MMBP$^{[H,L]}$) is a special case of the D-BMAP $^{[H, L]}$ in which at every discrete time there is at most one single arrival.

For the prioritized framework, the elementary network flow operators are defined as:

Output process

In a work-conserving queue, a cell is lost if and only if it finds the buffer full. Consequently, if we disregard the priority classification of the cells, we notice that the statistics of the output process of a D-BMAP$^{[H,L]}$/D/1/K queue are the same as the output process of a D-BMAP/D/1/K queue [13] (with no priority). We, therefore, compute the parameters for the output process in two steps (Figure 5). In the first step, we model the output process as a two state MMBP without taking into account the priority classification (as was done in section III)). In the second step, we compute the priority probability of a cell ($\vec{p}_{high}$).

Joining

We also take a two step approach when determining the D-BMAP[H,L] (process *c*) resulting from the superposition of two D-BMAP[H,L] (process a and b). We first compute the matrices $D_k^{(c)}$ and then the aggregate process priority probability, by taking into consideration not only the priority probability of each aggregating process but also their probability of arrivals. Thus, the i^{th} component of $p_{high}^{(c)}$ is given by:

$$p_{high}^{(c)}(i_c) = \sum_{j_c=1}^{M_c} \sum_{n_c=1}^{N_a+N_b} \sum_{n_a=min(0,\, n_c-N_b)}^{min(n_c,\, N_a)} H_a \times H_b \times \frac{z_a+z_b}{n_a+n_b} \times \left(d_{i_c j_c} \right)_{n_a,\, n_b}$$

$$H_a = \sum_{z_a=0}^{n_a} \binom{z_a}{n_a} \times p_{high}^{z_a}(i_a) \times p_{low}^{n_a-z_a}(i_a)$$

$$H_b = \sum_{z_b=0}^{n_b} \binom{z_b}{n_b} \times p_{high}^{z_b}(i_b) \times p_{low}^{n_b-z_b}(i_b)$$

where:

$n_c = n_a + n_b$

$p_{low}(i_a) = 1 - p_{high}(i_a)$

$p_{low}(i_b) = 1 - p_{high}(i_b)$

i_a and i_b are respectively the states of process *A* and *B* which correspond to state i_c

$\left(d_{i_c j_c}^{(c)} \right)_{n_a,\, n_b}$ is the element in the i_c^{th} row and the j_c^{th} column of $D_{n_a}^{(a)} \otimes D_{n_b}^{(b)}$

Splitting

The splitting operator is defined in the same way as the non-priority splitting operator. The priority probability seen by one destination has the same value as the priority probability before splitting.

VI)The Output Process of a D-BMAP$^{[H,L]}$/D/1/K Queue

Having already characterized the aggregate output process, we need to compute the probability that a cell belongs to a certain priority class. If we had an infinite buffer space the priority probability would be the same as the input process priority probability. However, in a finite buffer queue, we need to take into account the loss rate per class due to buffer overflow. We consider that the output process priority probability is independent of the state of the underlying Markov chain. Thus, our procedure is [13]:

$$\Pi_{high} = \frac{p_{high} \times (1 - R_{high})}{p_{high} \times (1 - R_{high}) + p_{low} \times (1 - R_{low})}$$

$$\Pi_{low} = \frac{p_{low} \times (1 - R_{low})}{p_{high} \times (1 - R_{high}) + p_{low} \times (1 - R_{low})}$$

where R_{high} (R_{low}) is the high (low) priority loss rate
Π_{high} (Π_{low}) is the output high (low) priority probability,
$p_{high} = \vec{p}_{high} \cdot \vec{\pi}$ ($p_{low} = \vec{p}_{low} \cdot \vec{\pi}$) is the input process high (low) priority probability

To compute the loss rates, we use a loss rate conservation law, which allows great reduction in the complexity of the solution. The conservation law establishes that the product of the aggregate loss rate times the aggregate arrival rate is equal to summation of the per class product of the loss rate times the arrival rate, i. e.:

$$\lambda R = \sum_{n=1}^{N} \lambda_n R_n$$

where λ and R are respectively the aggregate arrival rate and aggregate loss rate and λ_n and R_n are respectively the class n arrival rate and loss rate.

This loss rate law is a generalization of Clare and Rubin's loss probability conservation law for *i.i.d.* arrivals [29]. Their law establishes that the product of the aggregated

loss probability times the aggregate arrival rate equals the per class summation of the product of the loss probability times the arrival rate. Jeon and Viniotis [30] derived a similar law for queues with MMPP arrivals. Jeon and Vinioti‘s law states a relationship between the arrival rate and the loss rate conditioned on the state of the process at the beginning of busy periods. Although insightful, Jeon and Viniotis’ law has a limited applicability given that the related measures are not usual descriptors of a system. However, Clare and Rubin’s law cannot be applied to a queue with non-renewal arrivals. The main reason for this restriction is that in a non-renewal process we cannot relate time averages to steady-state statistical averages. In other words, the long term ratio between the number of losses and the number of cells does not converge to the definition of probability. Whenever we apply the concept of probability, we assume that we pick a random cell form the universe of cells and check if it will be lost or not. In a correlated process (non-renewal), the loss of a cell depends on the past loss history; it is not a random event. Actually, when trying to guarantee minimum Quality of Service, we are interested in the fraction of lost cells (loss rate) and not exactly in the loss of a particular (randomly selected) cell (loss probability). A proof of our loss rate conservation law can be found in [22].

To compute the loss rates, using the loss rate conservation law, we first solve the aggregated system by computing the queue length distribution. We then derive the low priority loss rate by observing a tagged low priority cell and computing the probability that it is not dropped (successfully transmitted). The high priority loss rate is computed by applying the conservation law. The solution of a discrete time queue is a straightforward generalization of the continuous time case [31].

To evaluate the accuracy of this priority computation procedure we use the same two node tandem network as in the non-priority case. To avoid any distortion of the percentage error, we limit the range of simulation results to those which can be obtained through Monte Carlo techniques (from 10^{-7} to 10^{-1}); therefore avoiding the use of rare event simulation (and consequently introducing another source of approximation).

Tables 3 and 4 show respectively the high and the low priority loss rate for a wide

range of values. Our procedure is more accurate when it estimates loss rate for the low priority class than it is for the high priority class. The errors of the low priority loss rate estimation are similar to the errors of the aggregated loss rate. We also notice that our procedure is more precise for high values of the loss rate than it is for lower ones. Errors were below 15% for the high priority class and below 10% for the low priority class.

In order to evaluate the impact of the offered load, its coefficient of variation and its correlation coefficient on the accuracy of the procedure, we vary respectively ρ, c and α. Figure 6 and 7 respectively show the high and the low priority loss rate as a function of ρ. Difference in the accuracy for both high and low priority class are under 5%.

Figure 8 displays the percentage error of the high and the low priority class when we vary c. We notice that our procedure gives more accurate results for higher values of the input process coefficient of variation. The impact of the coefficient of variation on the precision is more pronounced for the high priority class than for the low priority one. For the high priority class the maximum difference in the percentage error is 3% whereas for the low priority class the maximum difference is 2%. The accuracy increases significantly as p_{high} increases. This increase is more noticeable for the high priority class (under 7%) than for the low priority one (under 2%). In our validation experiments we also noticed that the precision of the procedure as a function of the coefficient of variance depends on the correlation coefficient. For positively correlated streams, we found out that the procedure is approximately 2% more precise than for negatively correlated streams. Regarding the correlation coefficient, the procedure is slightly more accurate for positively correlated streams than for negatively correlated ones (< 2%) (Figure 9)

To make sure that the interfering process parameters did not impact our results, we varied the interfering process ρ, c and α. No significant impact on the precision of our results was found. This remark is also valid for the buffer size from 50 to 200.

VII) Numerical Examples

In this section, we illustrate how our framework can be used to compute end-to-end performance in ATM virtual paths. To compute the end-to-end delay in an ATM virtual path, we make use of the parametric decomposition approximation, i.e., the queues are analyzed in isolation after their input process are fully characterized. In this approach, the dependencies among the queues are approximated by the flow parameters. We concentrate on ATM networks whose topology can be described as an acyclic directed graph. Otherwise, if we consider generally connected networks, we would have to define iterative procedures for determining the input flow of nodes in a cycle. We assume that there are two distinct sets of nodes: sets *E* and *I*. The elements of set *E* receive only input (external) traffic to the network (i.e., elements of set *E* are network's entry points). The elements of set *I* are nodes whose input is composed of the output process of other nodes and possibly input traffic to the network (i.e. nodes belonging to set *I* are network internal nodes which can also receive external traffic). We define S_k as the set of nodes whose input traffic can be determined only at iteration *k* of the computational procedure. In other words, nodes belonging to S_k have at least one input link whose flow parameters can only be computed at step *k-1*. We compute the occupancy distribution of all nodes of S_k at step *k*, and we denote a link whose traffic parameters have been determined as a marked link. The computational procedure can be summarized as [32]-[33]:

1 - $k = 1$ and $S_1 = E$;

2 - While $S_k \neq \varnothing$ do:

2.1 - Characterize the input process of every S_k node by performing a joining operation of all input links to each node. For S_1 nodes, the input processes are given by the input process to the network;

2.2 - Compute the steady state queue length distribution of every S_k node. Compute the mean delay seen by an arriving cell at an S_k node;

2.3 - Characterize the output process of every S_k node by matching the statistics of the output process with the statistics of a two-state MMBP;

2.4 - For each node in S_k , characterize the process of every outgoing link by performing a spliting operation output process;

2.5 - Mark all outgoing links of each node in S_k;

2.6 - $k = k + 1$.

By assuming a feed-forward topology, we guarantee that the computational procedure terminates.

We show two examples of feed-forward topology in Figures 10 and 11 [32]. The loads at nodes A, B and C are ($\rho = 0.75$, $c = 0.9$, $\alpha = 0.9$), ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$) ($\rho = 0.75$, $c = 0.1$, $\alpha = 0.9$) respectively. Table 3 presents the routing probabilities for the network of Figure 10. Table 4 and Table 5 respectively display the delay at each node and the end-to-end delays. For Figure 11 network, the arrival process parameters are ($\rho = 0.75$, $c = 0.9$, $\alpha = 0.9$) for nodes G, H, and J. The routing probabilities are given in Table 6. Table 7 and Table 8 show the delay at each node and the end-to-end delays respectively. We note that the end-to-end delay error estimation is in the range of the experiments described in section IV. Thus, the traffic mixing effect did not impact the precision of the results. To analyze the error trend as a function of the network size, we evaluate tandem networks with up to 20 nodes and found an error increase of less than 2% [22].

In Figure 12 we show an example of a four node tandem network where we vary the offered load to the first queue ($c = 0.9$ and $\alpha = 0.9$, $p_{high} = 0.8$). The interfering process parameters are the same for the three other queues ($\rho = 0.1$, $c = 0.5$, $\alpha = 0.9$, $p_{high} = 0.8$). We compute the end-to-end loss rate as $1 - \Pi (1 - p_i)$ where p_i is the loss rate at queue i. In the top part of Figure 12 we show the end-to-end loss rate computed using the approximate model and the simulation estimation. The bottom part of the figure shows the respective percentage error as in the single node case. We note that the precision increases as the offered load increases.

VIII) Conclusions

The dimensioning of the future ATM network demands appropriate queueing network models. In this paper, we introduced an approximate model for the output process of an

ATM multiplexer and extended this model to include selective discard mechanism. Moreover, we defined frameworks for queueing networks with Markov modulated flows. The approximate models are reasonably accurate. The percentage errors of the delay and loss rate estimation were under 7% and 10%. For the prioritized case, we found errors under 10% and 15% for the low and for the high priority class respectively.

This work can be extended in several directions. The investigation of queueing networks which take into consideration the representation of individual connections is useful for the derivation of per connection performances. Variations of the output procedure can be defined to compare the impact of different buffer organizations on the end-to-end loss performance.

We are currently investigating queueing networks in which the priority classification of the cells are correlated. We are also studying techniques for state space reduction for large scale networks and for modeling correlated splitting.

IX) References

[1]S. S. Lam and J. Wong, “Queueing netwok models of packet switching networks, Part 2: networks with population size constraints”N, *Perfor Eval.*, 2, pp 161-180, 1982.

[2]E. de Souza e Silva and R.R. Muntz, “Queueing Networks: Solutions and Applications”, in Stochastic Analysis of Computer and Communication Systems, H. Takagi editor, North Holland, 1990

[3]D. Mitra, J.A. Morrison and K.G. Ramakrishnan, “ATM Network Design and Optimization: A Multirate Loss Network Framework”, IEEE/ACM Transactions on Networking, vol. 4, no 4, pp 531-543, August 1996

[4]K.W. Ross, Multirate Loss Models for Broadband Telecommunications Networks, New York: Springer, 1995.

[5]F. Bonomi, S. Montagna and R. Paglino , “Busy period analysis for an ATM switching element output line”, in *Proc. of IEEE INFOCOM*, pp. 544-550, 1992.

[6]H. Saito, “The Departure process of an N/G/1 Queue”, *Perfor. Eval*., 11, pp. 241-251, 1990.

[7]G. Mayor and J.A. Silvester, “Time Scale Analysis of an ATM Queueing System with Long-Range Dependence Traffic”, to appear in *Proc of INFOCOM’97*, 1997.

[8]G. Mayor, J.A. Silvester and N.L.S. Fonseca, “Markov Modulated Approximation of Long-range Dependent Processes, unpublished manuscript, 1996.

[9]C. Blondia, “A discrete-time batch Markovian arrival process as B-ISDN traffic model”, *Belgian J. of Oper Res., Stat. and Comp. Science*, vol. 32 (3), pp. 3-23, 1992.

[10]C. Blondia and O. Casals, “Performance analysis of a Statistical multiplexing of VBR sources”, *Proc of IEEE INFOCOM*, pp 828-838, 1992.

[11]S. Wang and J. A. Silvester, “A discrete-time performance model for integrated service ATM multiplexers”, in *Proc. of IEEE GLOBECOM’93*, pp.757-761, 1993.

[12]N.L.S. Fonseca and J. A. Silvester, “Modelling the output process of an ATM multiplexer with Markov modulated arrivals”, in *Proc of IEEE ICC’94*, pp. 721-725.

[13]N. L. S. Fonseca and J. A. Silvester, “An approximate model for the output process of an ATM multiplexer with selective discard mechanism”, in *Proc of IEEE ICC’95*, pp 783-787.

[14]J.F. Ren, J. W. Mark and J. W. Wong, “End-to-end performance in ATM networks”, *Proc of IEEE ICC’94*, pp. 996-1002, 1994.

[15]V. Frost and Q. Wang, “Estimation of cell loss probabilities for tandem ATM queues”, *Proc of IEEE ICC’94*, pp. 1019-1024.

[16]W-C Lau and S-Q Li, “Traffic Analysis in Large-Scale High-Speed Integrated Networks: Validation of Nodal Decomposition Approach”, in *Proc. of INFOCOM’93*, pp 1320-1329.

[17]T. Takine, T. Suda and T. Hasegawa, “Cell loss and output process analyses of finite buffer discrete time queueing system with correlated arrivals”, in *Proc. of IEEE INFOCOM*, pp 1259-1268, 1993

[18]D. Park, H. G. Perros and H. Yamashita, “Approximate analysis of discrete-time tandem queueing networks with bursty and correlated input traffic and customers” to appear in Operation Research Letters.

[19]P. J. Kuehn, “Approximate analysis of general queueing networks by decomposition”, *IEEE Trans. Commun*., vol COM-27, 1, pp. 113-126, 1979.

[20] W. Whitt, “The Queueing Network Analyzer”, *The Bell Sys. Tech.l J*., vol. 62, pp. 2779-2815, Nov. 1983.

[21]I. Stavrakakis, “Efficient modeling of merging and spliting processes in large networking structures”, *IEEE J. Select. Areas Commun*., vol 9, no 8, pp. 1336-1347, Oct. 1991.

[22]N.L.S. Fonseca, “Queueing Network Models for Multiple Class Broadband Integrated Services Digital Networks”, U.S.C. CENG Tech Report 94-25, 1994.

[23]A. Y-M Lin and J. A. Silvester, “Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system, *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1524-1536, Dec. 1991

[24] Y. Le Boudec, “An efficient solution method for markov models of ATM links with loss priorities”, *IEEE J. Select. Areas Commun.*, vol. 9, pp. 408-417, Apr. 1991.

[25]H. Kroner, “Comparative performance study of space priority mechanisms for ATM networks”, in *Proc. IEEE INFOCOM’90*, pp. 1136-1143, 1990.

[26]J. Garcia and Olga Casals, “Stochastic models of space priority mechanisms with Markovian arrival processes”, Annals of Operation Research 35, pp. 271-296, 1992

[27]D. W. Petr and V. S. Frost, “Priority cell discarding for overload control in BIDN/ATM networks: an analysis framework”, International Journal of Digital and Analog Communication Systems, vol 3, n 2, 1990.

[28] N. L. S. Fonseca and J. A. Silvester, “A comparison of push-out policies in an ATM multiplexer”, in *Proc. of IEEE Pac. Rim Conf. on Commun. Comp. and Signal Proc*., pp. 338-341, 1993.

[29]L. P. Clare and I. Rubin, “Performance boundaries for prioritized multiplexing systems”, *IEEE Trans on Info. Theory*, n^{o} 3, pp. 329-340, May 1987.

[30]Y-H Jeon and I. Viniotis,”Achievable loss probabilities and buffer allocation policies in ATM nodes with correlated arrivals”, in *Proc of IEEE ICC’93*, pp. 352-358, 1993

[31]N. L. S. Fonseca and J. A. Silvester, “Estimating the loss probability in a multiplexer loaded with multi-priority MMPP streams”, in *Proc. IEEE ICC’93*, pp. 1037-1041, 1993.

[32]N. L. S. Fonseca and J. A. Silvester, “ On the computation of end-to-end delay in feed-forward ATM networks”, *Proc. of IEEE International Telecommunication Symposium’94,* pp. 460-464.

[33]J. A. Silvester, N. L. S. Fonseca and S. S. Wang,“D-BMAP Models for the Performance Analysis of ATM Networks”, em Performance Modelling of ATM Networks., D. Kouvatsos editor, Chapman and Hall Publishers, 1995

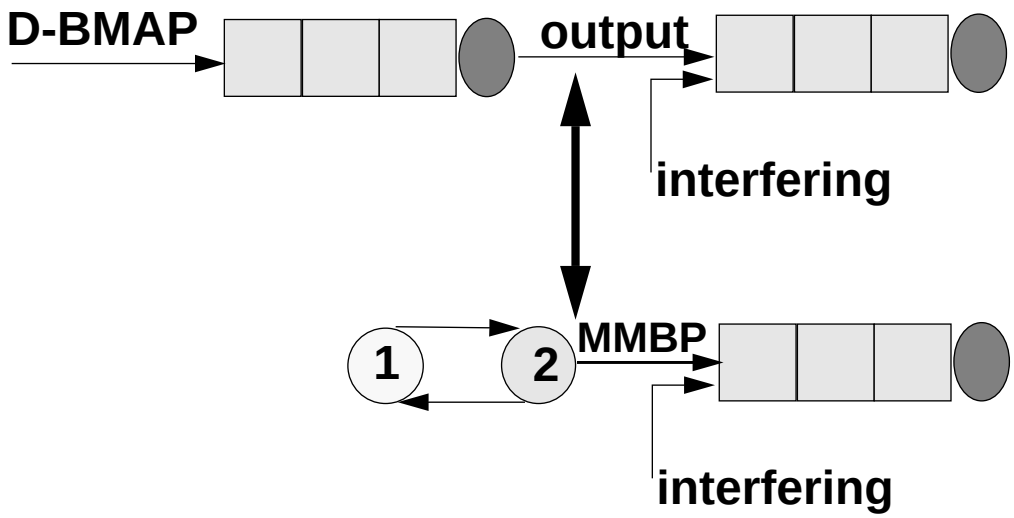

Figure 1: Scheme for the validation of the output procedure

Input (ρ, *c*, α)	Interfering (ρ, *c*, α)	Analytical	Simula	conf interval	error
(0.8, 0.3, 0.9)	(0.1, 0.1, 0.9)	6.11832	5.772	0.02	6.0
(0.8, 0.7, 0.9)	(0.1, 0.1, 0.9)	12.3299	11.676	0.07	5.6
(0.9, 0.7, 0.9)	(0.05, 0.1, 0.9)	21.787	20.809	0.03	4.7
(0.65, 0.9, 0.9)	(0.3, 0.5, 0.9)	30.6381	29.375	0.03	4.3
(0.75, 0.9, 0.7)	(0.24, 0.9, 0.9)	44.808	43.251	0.008	3.6
(0.9, 0.3, 0.9)	(0.1, 0.1, 0.9)	52.6774	51.044	0.06	3.2
(0.85, 0.9, 0.9)	(0.2, 0.5, 0.9)	73.8761	72.004	0.5	2.6
(0.9, 0.9, 0.9)	(0.2, 0.1, 0.9)	87.494	85.947	0.11	1.8
(0.9, 0.9, 0.9)	(0.3, 0.1, 0.9)	95.7508	94.522	0.18	1.3

Table 1: Delay at the second queue

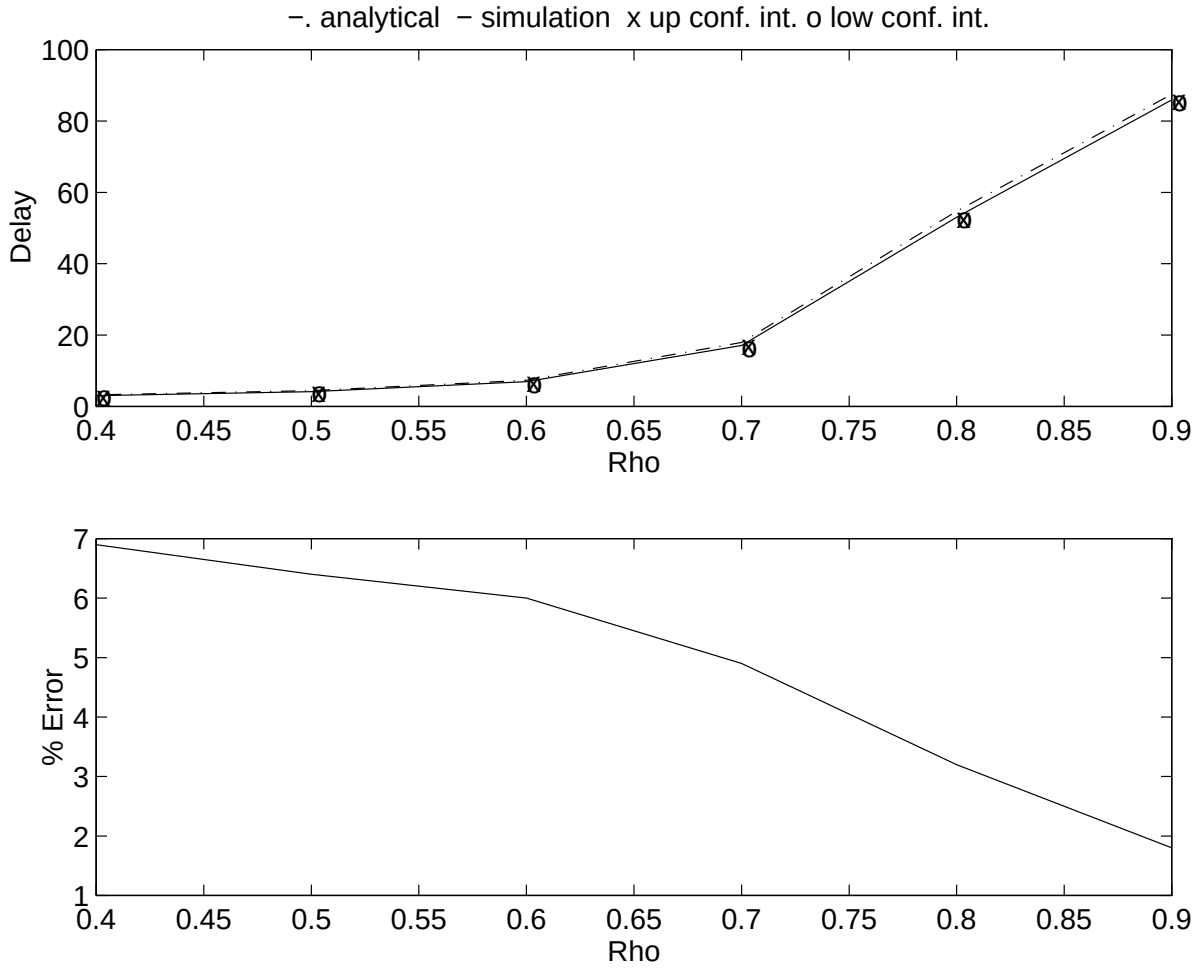

Figure 2: Delay estimation x ρ for (*c* = 0.9, α = 0.9) and interfering (ρ= 0.2, *c*= 0.1, α= 0.9)

input (ρ, *c)*	Analytical	simulation	conf interval	error
(0.8, 0.9)	1.3962e-1	1.3047e-1	2.35e-3	4.2
(0.75, 0.9)	2.98797e-2	2.8322e-2	6.38e-4	5.5
(0.7, 0.9)	2.5675e-3	2.4245e-3	5.18e-5	5.2
(0.675, 0.9)	6.01925e-4	5.6839e-4	4.93e-6	5.9
(0.65, 0.9)	8.4937e-05	8.0281e-05	1.82e-07	6.8
(0.75, 0.1)	5.2704e-06	4.8397e-06	2.94e-08	8.9
(0.7, 0.47)	3.0113e-08	2.7779-e-07	5.10e-09	9.4

Table 2: Loss rate at the second queue for input $\alpha = 0.9$ and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$)

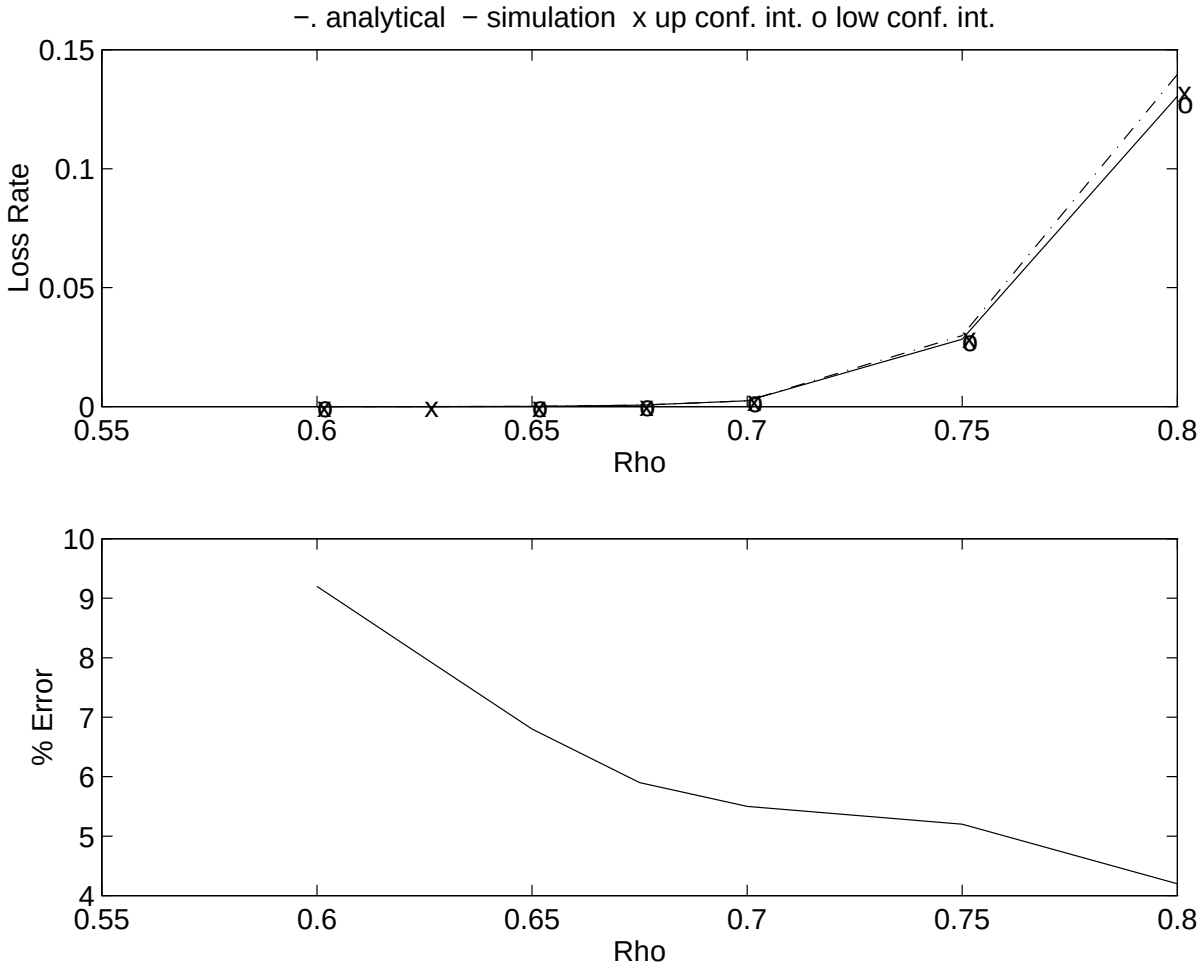

Figure 3: Loss rate estimation x ρ (*c* = 0.9, α= 0.9) and interfering (ρ= 0.2, *c*= 0.1, α= 0.9)

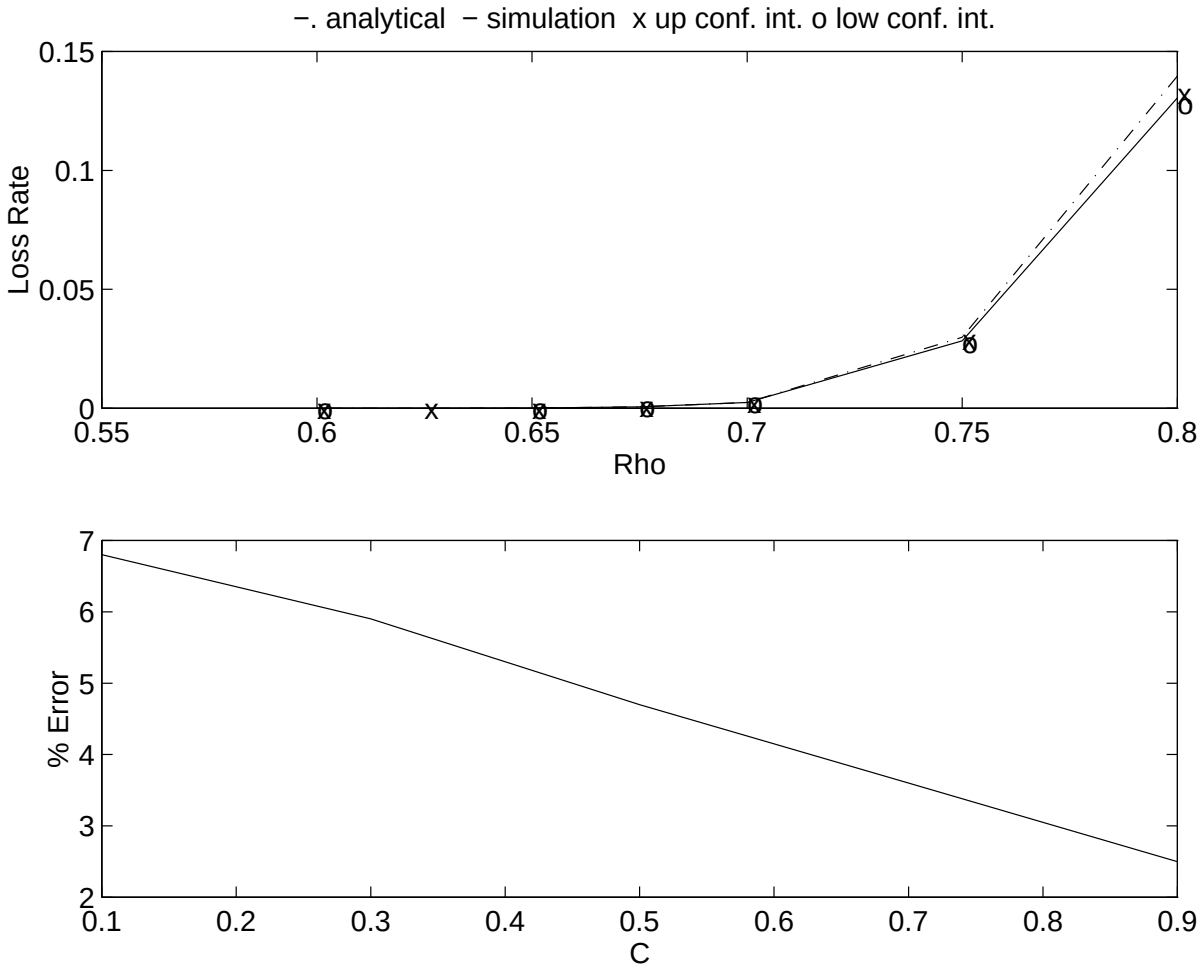

Figure 4:Loss rate estimation x *c* (ρ=0.75, α= 0.9) and interfering (ρ= 0.2, *c*= 0.1, α= 0.9)

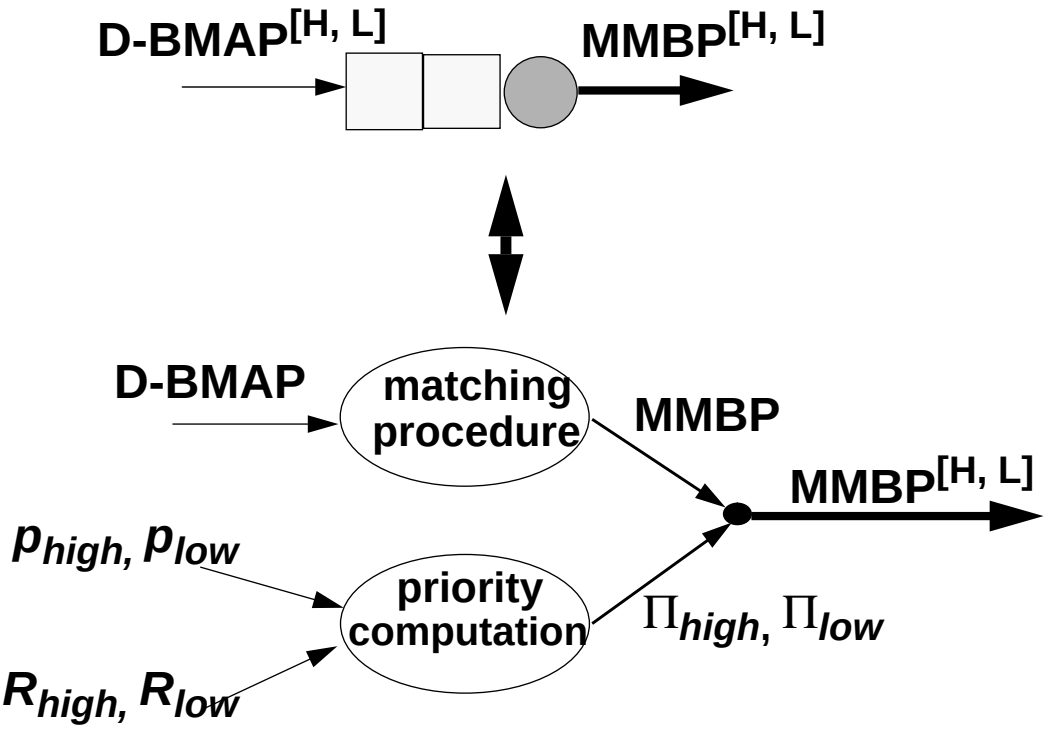

Figure 5: A procedure for modelling the output process of a multiplexer with selective discard mechanism

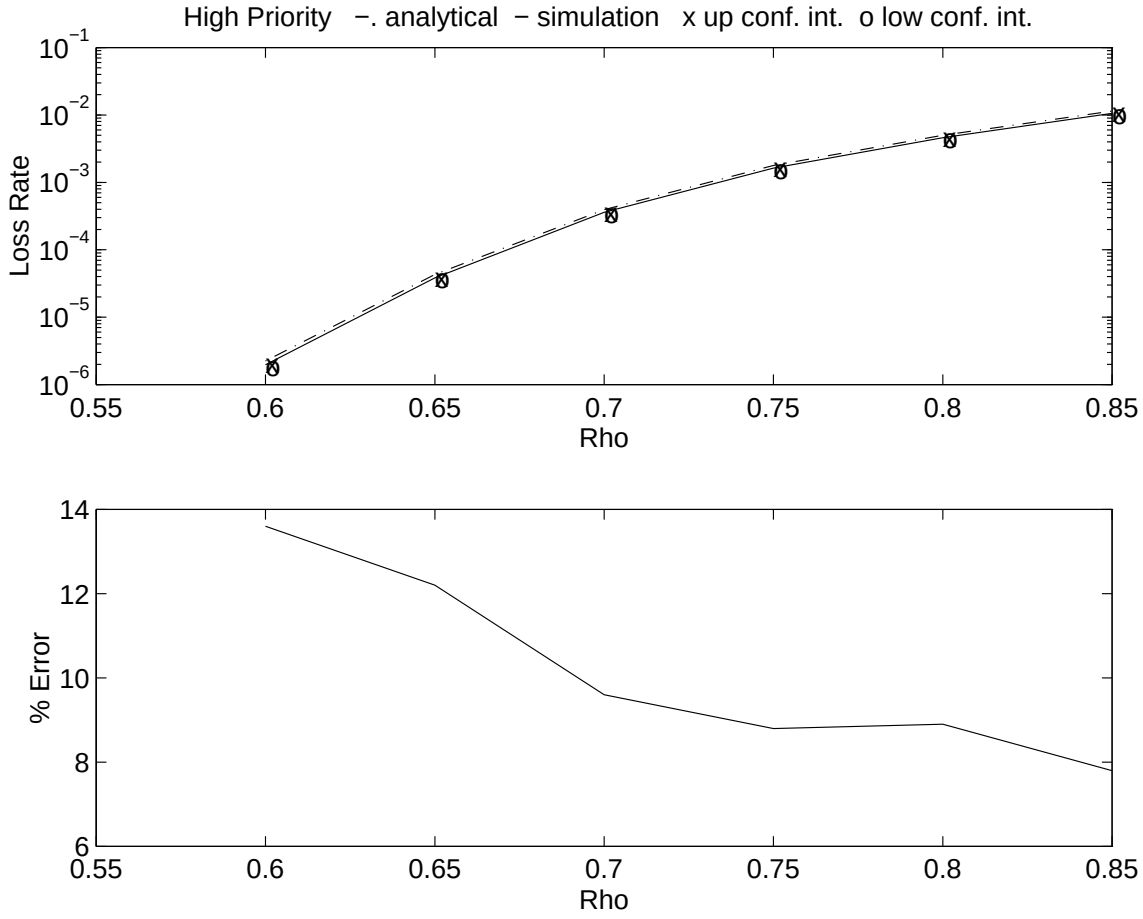

Figure 6: High Priority Loss rate estimation x ρ for (c=0.9, α=0.9, p_{high}=0.8) and interfering (ρ=0.5, c=0.1, a=0.1, p_{high}=0.7)

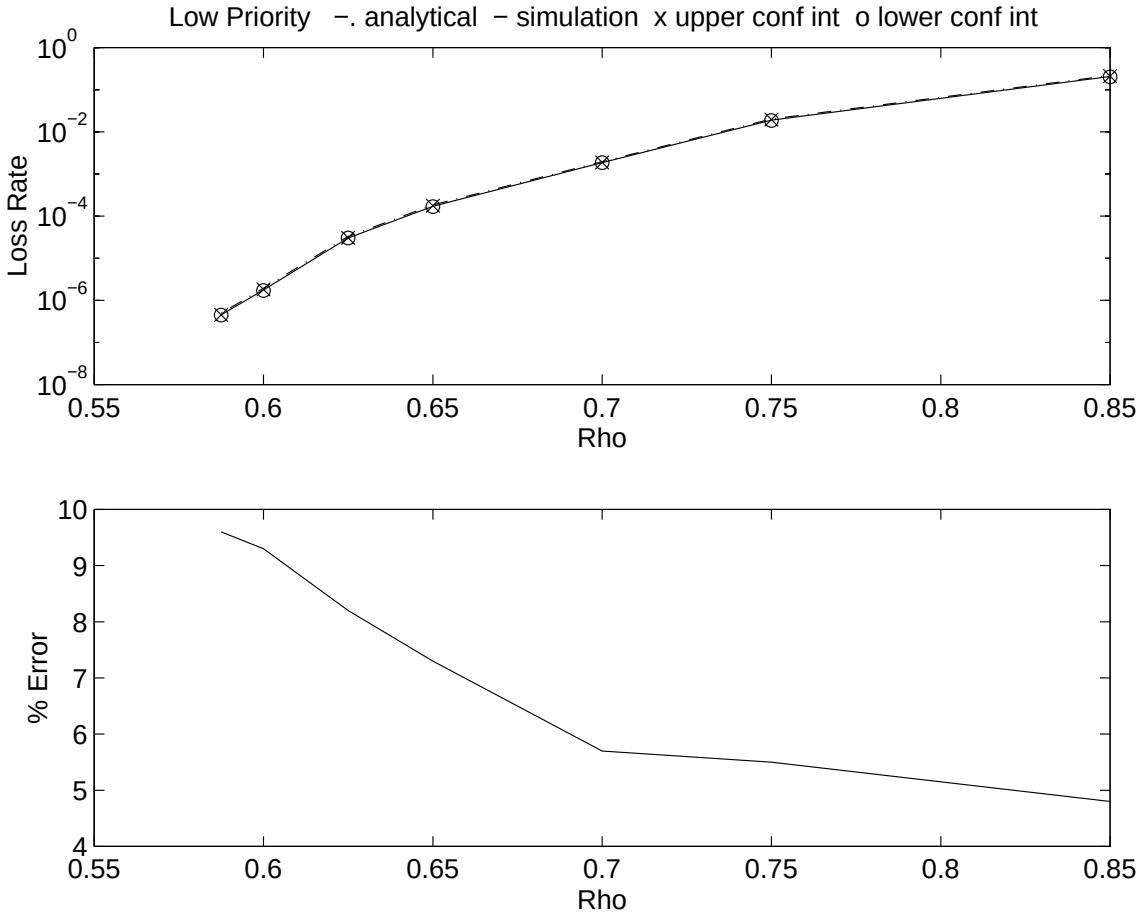

Figure 7: Low priority loss rate x ρ for (c=0.9, α=0.9, p_{high}=0.8) and interfering (ρ=0.5, c=0.1, a=0.1, p_{high}=0.7)

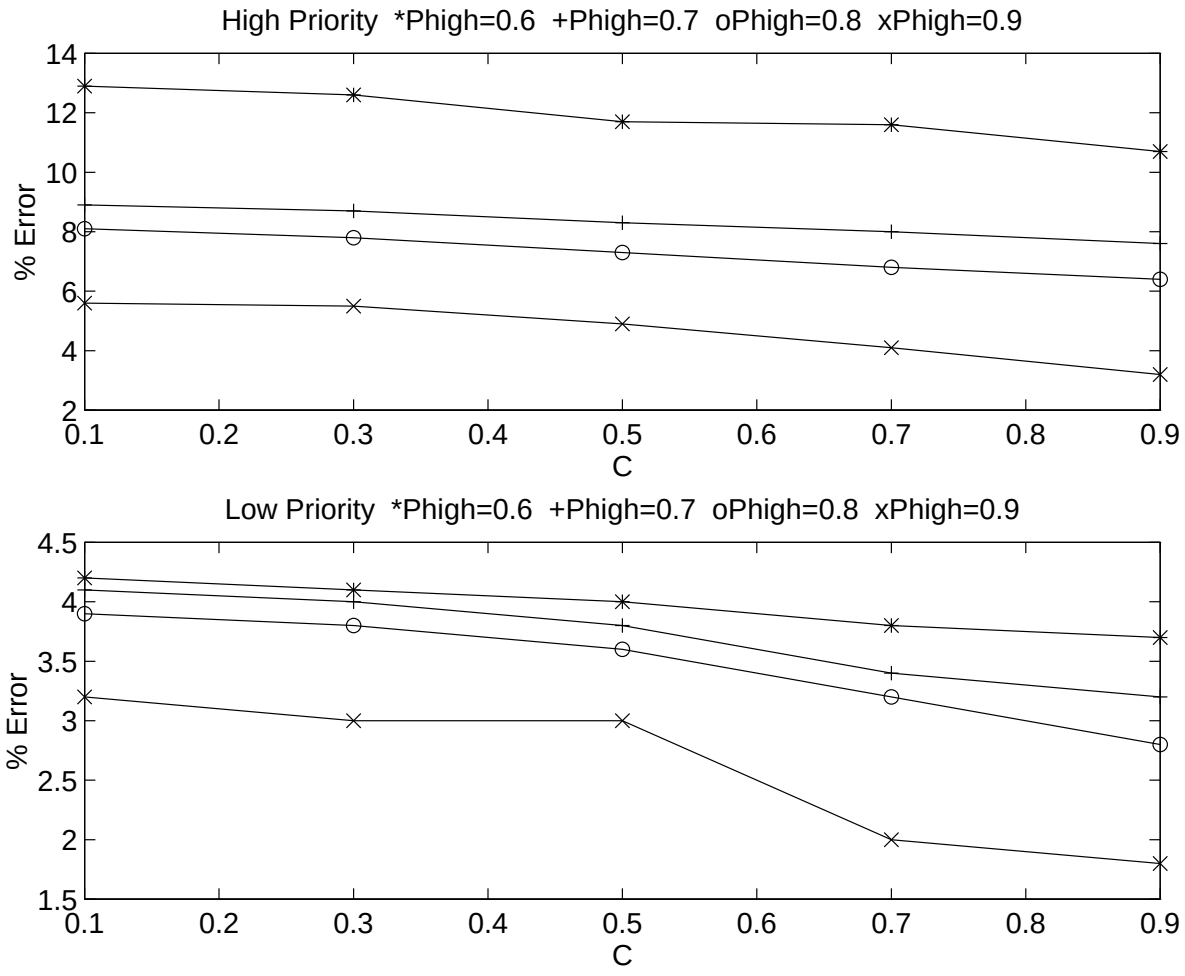

Figure 8: Loss rate estimation x *c* for (ρ= 0.8, α=0.1) and interfering (ρ= 0.5, *c* =0.1, α=0.1, p_{high}=0.7)

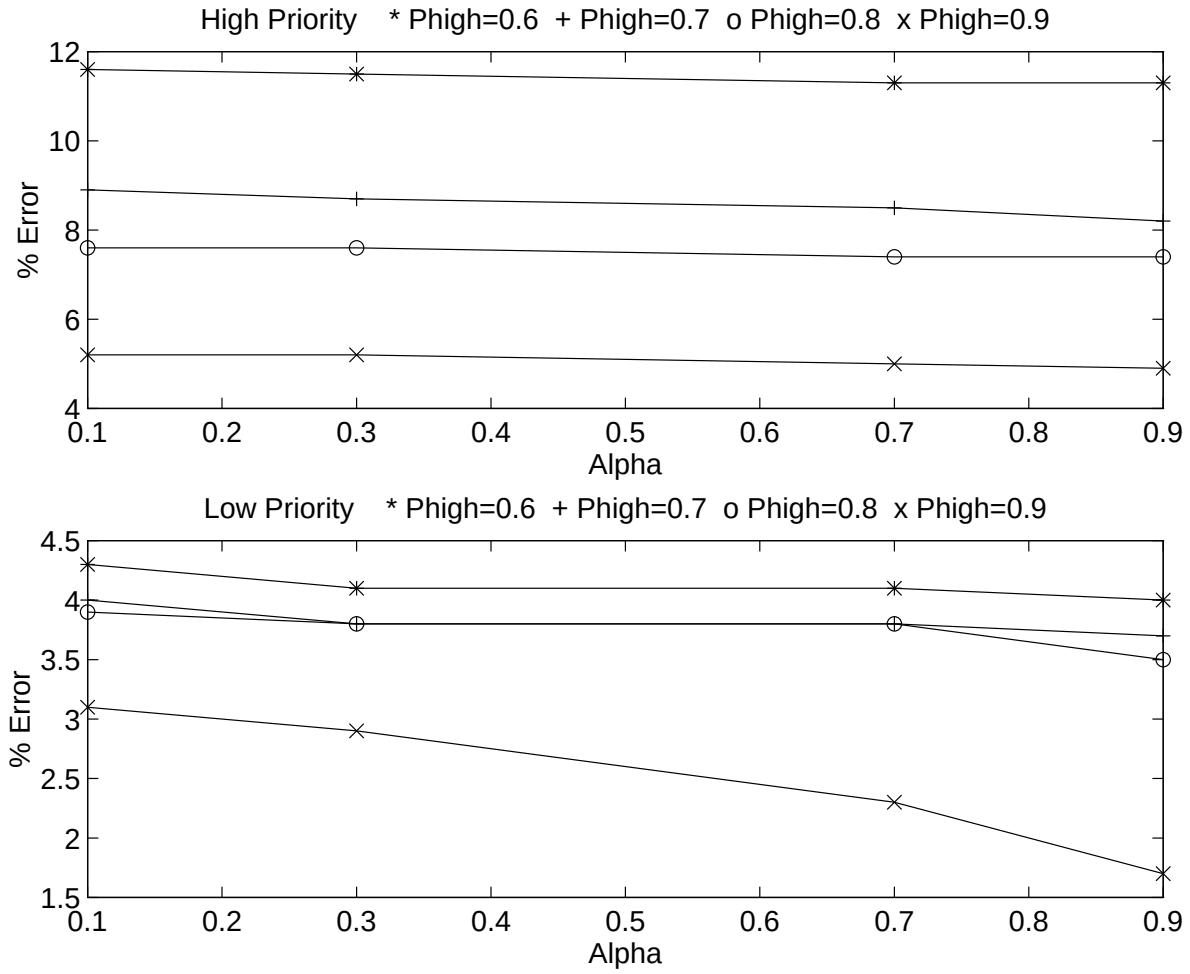

Figure 9: Loss rate estimation x α for (ρ= 0.8, *c*= 0.1) and interfering (ρ= 0.5, *c*= 0.1, α= 0.9, p_{high} = 0.7)

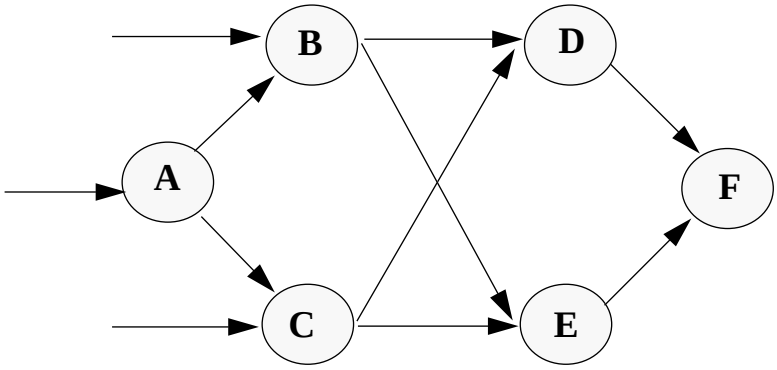

Figure 10: First feed-forward network

	B	C	D	E
A	0.7	0.3		
B			0.4	0.6
C			0.6	0.4

Table 3: Routing probabilities for figure 14 network

	estimated	simulation	error
A	11.73	11.01 ± 0.16	6.5
B	69.38	66.65 ± 0.65	4.1
C	26.81	25.49 ± 0.43	5.2
D	22.68	21.42 ± 0.11	5.9
E	25.91	24.56 ± 0.10	5.5

Table 4: Delay per node for figure 14 network

	estimated	simulation	error
ABD	103.79	99.08	4.7
ABE	107.08	102.22	4.9
ACD	61.22	57.92	5.7
ACE	64.45	61.06	5.5

Table 5: End-to-end delays for Figure 14 network

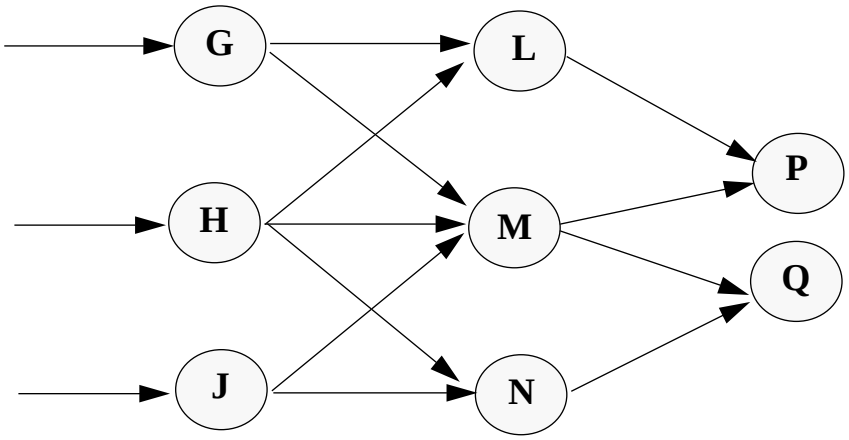

Figure 11: Second Feed-forward network

	L	M	N	P	Q
G	0.9	0.1			
H	0.1	0.8	0.1		
J		0.3	0.7		
M				0.5	0.5

Table 6: Routing probabilities Figure 15 network

	estimated	simulation	error
G	11.64	11.04 ± 0.08	5.4
H	11.64	11.00 ± 0.07	5.8
J	11.64	11.05 ± 0.06	5.3
L	2.62	2.46 ± 0.02	6.5
M	12.97	12.34 ± 0.12	5.1
N	2.52	2.36 ± 0.01	6.8
P	48.07	46.08 ± 0.78	4.2
Q	47.89	45.83 ± 0.61	4.5

Table 7: Delay per node Figure 15 network

	estimated	simulation	error
GLP	62.33	59.57	4.6
GMP	72.68	69.45	4.7
GMQ	72.5	69.21	4.8
HLP	62.33	59.59	4.6
HMP	72.68	69.42	4.7
HMQ	72.5	69.17	4.8
HNQ	62.05	59.44	4.4
JMP	72.68	69.47	4.6
JMQ	72.5	69.22	4.7
JNQ	62.05	59.24	4.7

Table 8: End-to-end delays for Figure 15 network

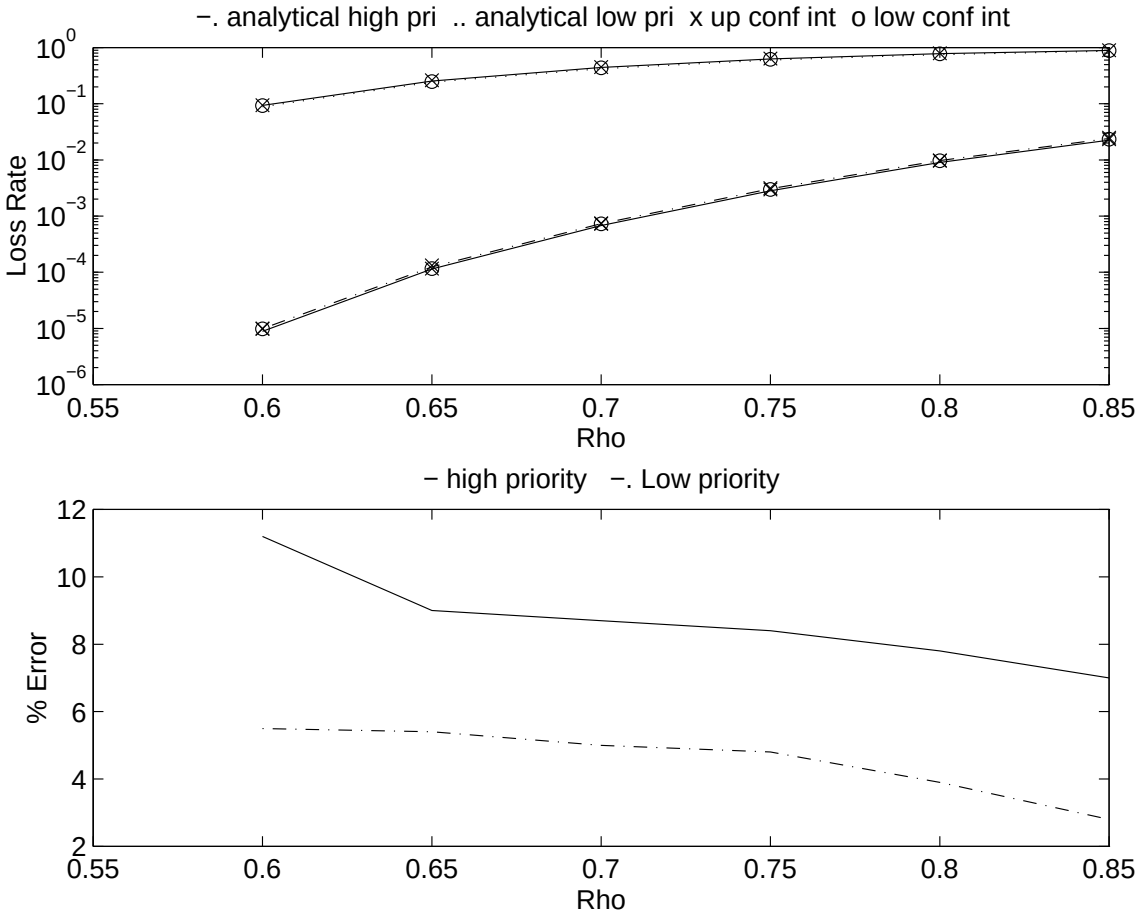

Figura 12: Loss rate estimation x r for a tandem network with 4 nodes and with (c= 0.9, α= 0.9, p_{high}= 0.8) and interfering (ρ=0.1 , c= 0.5, α= 0.9, p_{high}=0.8)