TECHNICAL ARTICLE

SIMULATION 68:1, 009-022
© 1997, Simulation Councils, Inc.
ISSN 0037-5497/97 $3.00 + .10
Printed in the United States of America

Two Simulation Tools for Testing ATM Resource Allocation Strategies

Raffaele Bolla, Ahmad Dalal'ah, Franco Davoli, Mario Marchese
Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Genoa, Italy
franco@dist.unige.it

Two event-driven simulation tools aimed at testing access control and routing mechanisms in an Asynchronous Transfer Mode (ATM) environment are presented. The first one is dedicated to the behavioural description of an ATM Virtual Circuit (VC) switch. The purpose is to test resource allocation and Call Admission Control (CAC) strategies, so only the relevant events for this objective are considered and some important functions (e.g., the switching element) for a complete description of an ATM switch are not explicitly modeled. The second simulator has been designed to test routing strategies for an ATM network. In such a case, a higher level of abstraction than in the previous one is necessary and, again, only the meaningful events to the aim are taken into account. Some resource allocation, CAC, and routing schemes are also reported, along with the description of the simulation tools. Several simulation results are discussed, in order to assess their performance.

Keywords: Discrete-event simulation, ATM, routing

1. Introduction

Due to the increasing complexity of telecommunication systems and networks and to the large number of technical details to be accounted for, simulation tools have reached considerable importance for performance analysis in this area. Analytical modeling of data transmission details, through a computer network (or, more generally, of services in an integrated services network), is often a very difficult task, because it may involve many functionalities throughout a layered architecture: from application layer protocols, which are, substantially, software procedures, to the hardware devices used for transmission over the physical channel. On the other hand, the implementation of a prototype for network analysis, with a research aim, is complex and expensive. Both universities and research and development laboratories inside a company often prefer the utilization of simulation tools to investigate the behaviour of the system under analysis. Another advantage of simulation is that a small part of a whole system can be picked out, if of interest, and other complex details, even though essential for a real implementation, can be temporarily not considered.

The specific application area of high-speed networks treated in this paper has received a great deal of attention in these last few years. In more detail, ATM networking has been the object of research from many points of view: physical transmission [1], switching [2], operation and maintenance [3], and CAC and routing at the call level. As far as the last two topics are concerned, a huge amount of literature is available: [4–8], for example, are dedicated to call admission schemes, while [9–18] refer to bandwidth allocation in order to guarantee Quality of Service (QoS). Concerning routing, [19–26] is a list of references on the topic. Most of the papers listed above use simulation to analyze the performances of their proposals, but they do not describe the structure of the simulation tools. On the contrary, [27] and [28] explicitly indicate some details about the utilized ATM simulator; and [29–32] provide some information about simulation methodologies, although in a different network context. Reference [33] contains a chapter on the fundamentals of ATM simulation.

Two simulators will be presented in this paper. The first one has been used to test the efficiency of the CAC schemes in [5] [8] and [9]; the second one to analyze the routing policies in [19]; so, the objective is not the design of an ATM switch general purpose simulator, but the description of tools that explicitly suit CAC and routing in a particular environment. Event-driven simulators have been designed in both cases, because, as explicitly indicated in the following, the systems under analysis can be modeled as a sequence of countable events, and nothing of interest is supposed to take place between the mentioned events [34]. Simulated time is advanced by using *next-event* time advance [35]; time is moved from the instant of the current event to the time of the next scheduled one. The random variable generation has been performed by following the indications of the literature [36, 37], as well as the decisions about the simulation length [38].

In more detail, the quantities needed to be measured here are, concerning the node simulator, the cell loss rate (i.e., the ratio between the cells arrived with a full buffer and the total number of arrived cells) and the queuing delay (the average time necessary to traverse the buffer); as far as the network simulator is concerned, the number of blocked calls and the overall number of call requests are the meaningful quantities. The traffic is divided into classes (by assuming Service Separation, as discussed in [39]), which are characterized by performance requirements as packet loss rate and delayed cell rate, and by statistical parameters as peak and average bandwidth. The switching element considered is an output buffer one, where a buffer of fixed length is dedicated to each traffic class at each outgoing link. The number of input links is limited only by the number of accepted connections, as is often done in the literature [27–29]. Particular attention is dedicated to the modelling of the traffic behaviour: each connection is considered independent of the others and a Talkspurt-Silence model has been chosen to represent each call. This model is particularly suitable to describe bursty traffic for voice and data [40] and has been extensively used as a traffic generator in other studies; [41], for example.

The paper is structured as follows. Section 2 is dedicated to the ATM node simulator, and is divided into subsections, which describe: the traffic generators, a general behavioural description of the ATM node, the events used in the simulator, and the different resource allocation and CAC schemes utilized in the results. The presentation of the network simulator and of the performed routing strategy is the object of Section 3, also organized in subsections. Simulation results in both cases are shown in Section 4. Section 5 contains the conclusions.

2. The ATM Switch Simulator

2.1 *Traffic Generation*

As already mentioned, the traffic is divided into a number H of classes, and each connection is considered statistically independent of the others and described by a Talkspurt-Silence model, which is a two-state Markov chain alternating periods of activity (burst periods) and periods of silence. Within a burst period cells are generated with known and constant bit rate, $B_p^{(h)}$ (peak rate). If the model is considered in the discrete-time domain, the sojourn times in both states are geometrically distributed with known mean value, $B^{(h)}$ (average burst length). The quantity $B_a^{(h)}$ is the average bit rate of a call; the ratio between the peak and average bit rate ($b^{(h)} = B_p^{(h)} / B_a^{(h)}$) is called burstiness and is used as a measure of the *bursty activity* of a source.

The traffic source behaviour is depicted in Figure 1, where three different levels are taken into account: the connection (or call) level (whose duration is exponentially distributed with average value $1/\mu^{(h)}$), the burst level, and the cell level, whose characteristics have been described above. In the case shown, the peak rate of the call is one fourth of the total capacity of the channel, as can be noted in the cell

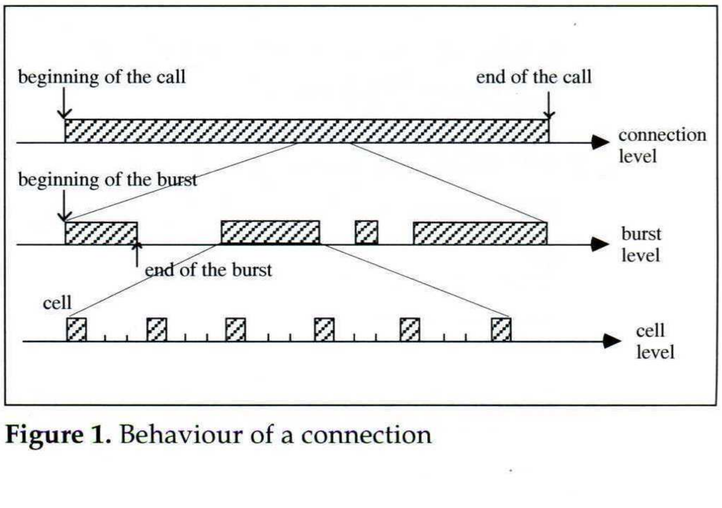

Figure 1. Behaviour of a connection

level in Figure 1, where there are four slots between two consecutive cell generations.

Concerning the connection request arrival process, a Poissonian distribution has been chosen. It can be observed that the description of a connection is maintained to a level high enough to bypass many functions which are not meaningful in this context; e.g., the physical layer, and both the Physical Medium Dependent (PMD) and the Transmission Control (TC) sublayers.

2.2 *Model of the ATM Switch*

Even the VC switch model is extremely focused on particular quantities, relevant to the aim. The switching element is completely bypassed: any hardware component dedicated to the reception or the processing (e.g., CRC error codes) of the ATM cells is not considered, as well as the physical elements for the generation, adaptation, and recovery of the transmission frame. In this approach, the node is simply a set of queues of fixed length. The cells are picked out by a scheduler following an algorithm ruled by the switch control element, independent of the simulation tool implementation. Some examples of control elements can be found in Section 2.4.

The "abstraction" used of an ATM switch is shown in Figure 2. Each cell belonging to a connection of traffic class h, generated as indicated in the previous subsection, enters a buffer dedicated to that traffic class. Even if apparently not so detailed, the description is accurate enough to test a CAC scheme. The simulator, in fact, can compute the number of lost cells and the queuing delays, which are the quantities of interest in this context. Only one output channel (single ATM multiplexer) is considered for the sake of simplicity; this simplification will be removed in the routing case.

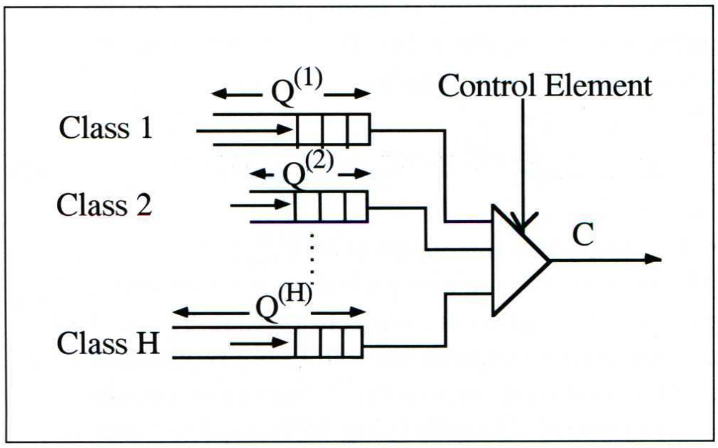

Figure 2. Model of the ATM switch

2.3 *Event Scheduling*

An event-driven simulator has been designed. Time is moved on by the events themselves. The tool has been implemented in C language, considering the following meaningful events: *connection_request; init_burst; arrival_cell; enqueue_cell; transmission;* and *connection_end.*

The event *connection_request* identifies a generic class-h call arrival; the acceptance or rejection of a call is decided (by interrogating the control part by means of a software routine) following a certain rule; see Section 2.4. If the connection is accepted, the events *init_burst* and *connection_end* are scheduled for that connection; *init_burst* is immediately scheduled (i.e., after 0.0 s), whereas a random variable with average value $1/\mu^{(h)}$ s is the scheduling time, concerning *connection_end*. Furthermore, the number of connections in progress (a simple counter) is incremented. On the other hand, if the connection cannot be accepted, the number of rejected connections is increased. In both cases, the next *connection_request* is always scheduled with time depending on the decided average traffic load. As mentioned, the arrival statistics are Poissonian, and their average value determines the load.

Init_burst is the beginning of the burst phase. The burst duration is decided, and the arrival of the first cell (*arrival_cell*) is scheduled by following the peak rate of the class to which the connection belongs. The next burst is scheduled when the current one is finished; the scheduling time is a geometric random variable, simply computed from the Markov chain [36] of the Talkspurt-Silence model.

Arrival_cell increases the total number of arrived cells, requires to enqueue the cell arrived (*enqueue_cell*) and reschedules itself (i.e., *arrival_cell*) for the duration of the burst.

The event *enqueue_cell* works as follows: if any room is left, the cell is queued, the number of cells in the queue is incremented, and the input time is marked. Otherwise, if the queue is full, the cell is discarded and the number of lost cells is increased.

Transmission is scheduled with time T_s (time slot duration, depending on the channel capacity) in the initial phase of the simulation; then, the next event of this type is automatically scheduled with the same time at each *transmission*. The effect is to have a temporal division at the slot level, without loading the data structure on which the simulator is based. The cells are picked out from the proper buffer and physically transmitted. The "output scheduling" mechanism, as mentioned above, depends on the Control Element, which indicates which buffer has to be served; even in this case the control mechanism is interrogated by a simple routine.

Connection_end is the last event listed. The number of ended connections is updated here.

This set of events is sufficient to reach the desired level of accuracy, and it does not vary by changing the admission control algorithm; i.e., this part is independent of the CAC and resource allocation scheme.

Both the CAC scheme and the mechanism used by the "output scheduler" are strictly related to the bandwidth allocation mechanism, i.e., how the bandwidth resource is allocated to the different traffic classes. So, if resources are not off-line statically allocated and a dynamic bandwidth allocation strategy has to be tested, an event, called *resource_allocation* in the following, is necessary. The temporal scheduling of this event depends on the algorithm used. If the scheme is completely dynamic, the event is not needed because it can be substituted by a software routine called again at each *connection_request*. In the next section a periodic reallocation mechanism is introduced to give an example of a possible application of the simulator.

2.4 *Bandwidth Allocation and CAC Schemes*

The CAC and bandwidth allocation controllers work as follows. An admission controller implements a decision rule for the acceptance of incoming calls that depends on the current number of accepted connections for the specific class, on the statistical and performance characteristics of the connections, and on the bandwidth (termed "virtual capacity" in the following), which has been assigned for the current period by the allocation controller. The latter divides the total capacity C of the ATM channel [Mbits/s] into virtual capacities $V_m^{(h)}$, h = 1,..., M, where m = 0, K, 2K,... represent the instants of intervention (scheduling time of the event *resource_allocation*), and K is the length of the intervention period (in slots); the m-th assignment holds constant for the time slots k = m, m + 1,..., m + K – 1. The assignment is done on the basis of the dynamic variations in the traffic flows, with the goal of providing a fair sharing to the admission controller. To this aim, a suitable cost function that takes into account the expected number of lost cells pertaining to the whole offered traffic over the following K slots, is minimized at each instant m, where a new K-slot period begins.

In the model that we will use to derive the access control rules and the cost function, we suppose the slot assignment to be such that class h can see an average service capacity, $V_m^{(h)}$, independently of the actual utilization of the channel. This would imply that the available channel slots be distributed among the traffic classes in proportions that are in accordance with the $V_m^{(h)}$'s, even if a certain class has temporarily not enough flow to fill its assigned slots and some waste may be created.

So we must have

$$\sum_{h=1}^{M} V_m^{(h)} = C \qquad m = 0, K, 2K,\ldots \tag{1}$$

In the analytical derivation which is needed to evaluate the lost and delayed cell probabilities that build up the cost function used in the reallocation, the nature of class-h traffic is supposed to be made up by bursty connections with statistical traffic characteristics that are identical and independent of each other; this situation is the same as the simulation case. Each bursty connection is represented by means of an Interrupted Bernoulli Process (IBP) [34]. Let $N^{(h)}$ be a given number of multiplexed connections of a class h; as they are independent of each other, the steady state probability $v_{N^{(h)},n^{(h)}}$ of having only $n^{(h)}$ active connections out of $N^{(h)}$ accepted connections is given by a simple binomial distribution.

The CAC scheme is embedded in a software routine managed by the event *connection_request*.

2.4.1 *Admission Control.* The acceptance algorithm that we use is based on two controls: satisfying loss and delay requirements, respectively. As far as the first control is concerned, we impose an upper limit $\varepsilon^{(h)}$ on the value of the cell loss rate, namely

$$\sum_{n=0}^{N^{(h)}} P_{\text{loss}}^{(h)}(n) v_{N^{(h)},n} \le \varepsilon^{(h)} \tag{2}$$

where $P_{\text{loss}}^{(h)}(n)$ represents the steady-state value of the cell loss rate, conditional on n. Its expression is not reported here for the sake of simplicity; it can be found in [5]. From inequality 2 we can obtain the maximum number of connections, $N_{\max,L}^{(h)}(m)$, that $V_m^{(h)}$ and $Q^{(h)}$ can support as regards the cell loss requirement. Concerning the delay constraint, it is imposed by requiring that the probability of the cell delay exceeding the value $D^{(h)}$ (in time slots) be lower than a given threshold $\delta^{(h)}$, namely

$$\sum_{n=0}^{N^{(h)}} P_{\text{delay}}^{(h)}(n) v_{N^{(h)},n} \le \delta^{(h)} . \tag{3}$$

The complete formulation of $P_{\text{delay}}^{(h)}(n)$ is given in [9]. As done for the loss probability, the maximum number $N_{\max,D}^{(h)}(m)$ of connections that $V_m^{(h)}$ and $Q^{(h)}$ can support as regards the cell delay requirement can be obtained from inequality 3. Summing up, the access control rule satisfying both requirements is simply the following: a new connection of the h-th class that arrives at time slot k, $m \leq k \leq m + K - 1$, can be accepted in the network if

$$N_A^{(h)}(k) + 1 \leq \min\left[N_{\max,D}^{(h)}(m),\ N_{\max,L}^{(h)}(m)\right] \equiv N_{\max}^{(h)}(m) \qquad (4)$$

$N_A^{(h)}(k)$ being the number of connections of the h-th class in progress at slot k. Note that the admission control rule in this case does not require any particular calculation.

2.4.2 *Bandwidth Reassignment.* As regards the bandwidth allocation, at each decision instant m, the virtual capacities $V_m^{(h)}$ are dynamically reassigned by the allocation controller by means of a process that minimizes a suitable cost function J_p. The function to be minimized [5] is based on the knowledge of the traffic intensities of the various classes, observed over the previous period, and forecasts the total number of cells that would be lost under that load in the current period. The aim is that of compensating sudden load variations in some of the classes, by varying their share of the channel accordingly.

In the minimization of the cost function J_p, account must be taken of the equality constraint 1, as well as of the inequality constraints

$$V_m^{(h)} \geq V_{\min}^{(h)}(m) \qquad (5)$$

which serve the purpose of ensuring service quality to the $N_A^{(h)}(m)$ connections already in progress. Actually, $V_{\min}^{(h)}(m)$ is the minimum capacity that is necessary to cope with these connections, and can be computed so as to satisfy both cell loss and delay requirements. The minimization of the cost function under constraints 1 and 5 is a mathematical programming problem that can be performed by means of a gradient projection method.

As already said, this last part is embedded in the event *resource_allocation*, which can be changed if a different strategy has to be tested. Actually, the simulator has been designed just to test different algorithms of this type, and the presentation of Sections 2.4.1 and 2.4.2 can be considered as an example of application.

3. The ATM Network Simulator

In this case, an extension of the previous model, which is referred to as a *single-node, single-output link*, has to be considered to take into account more than one outgoing link at each node. Concerning the analytical treatment of the routing problem, an index ij has to be added to each of the quantities defined in the previous section. As far as the simulator is concerned, only the event *resource_allocation* and the routine managing the CAC scheme have to be modified to cover this new situation. A simple parameter, indicating the link chosen, is sufficient to solve the problem.

3.1 *Network Model*

The most important "object" to be described in this context is the interconnection among the nodes, i.e., the network topology. In the simulator, the "network" is only a "matrix," whose elements contain important information for each link. The network model is very simple due to the relatively high level of abstraction. The physical connection between two nodes is identified by the parameter "connected" appearing in each matrix element, which can assume the value "1," if there is direct connection, and "0," otherwise. A simple six-node network is depicted in Figure 3, along with its abstract view. It can be noted that this model takes into account the direction of the links; for example, node 1 is connected with node 2, but not vice versa.

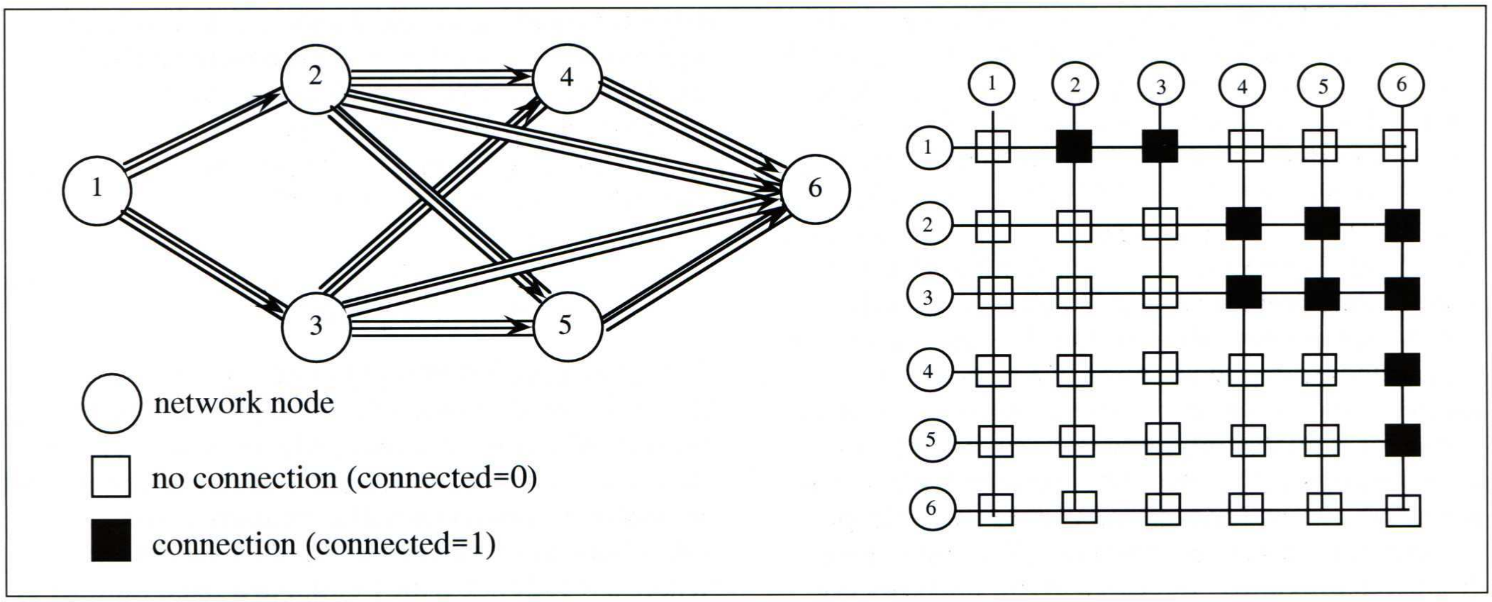

Figure 3. Network model

The matrix is replicated for each traffic class, because the quantities considered have a meaning only on a class basis, and Service Separation decouples the network into sub-networks. The "cell level" part, very important to the node, is bypassed here because the efficiency of the CAC scheme has been already evaluated. Only the routing strategy is evaluated.

3.2 *Event Scheduling*

Also in this case, the list of events is very limited because, in this context, the network performance is measured in terms of the rate of accepted and blocked connections. Thus, the main events needed to be considered are: *connection_request* and *connection_end*.

connection_request contains the decision about the acceptance or rejection of a call (as in Section 2.3), and the choice of the best path to get to the selected destination. The best path is decided step-by-step, by rescheduling the event *connection_request* at each node chosen, until the destination is reached; the complete routing procedure is described in the next subsection. The implementation of the strategy is embedded in a software routine. If a connection is accepted (i.e., the destination has been reached and a source-destination route has been found), the event *connection_end* is scheduled. Otherwise, a routine to free the resources already reserved at the previously visited nodes is necessary. Then, the next *connection_request* is scheduled.

connection_end contains only the upgrading of some quantities (namely, the costs defined in the next section) and of the resource status. This operation is performed by using software routines and, as in the previous case, cannot be considered as an actual part of the simulator.

Two other events are *resource_allocation* (as above) and *cost_communication*. The latter will be clarified in the next section, since it is strictly dependent on the routing algorithm. Due to the high level of abstraction, the network simulator is very fast and hours of network behaviour can be simulated in a few seconds.

3.3 *Routing Strategy*

The routing strategy (called DLCP in [19]) works as follows. At connection setup, a call request packet is forwarded from node to node in a hop-by-hop fashion. At each VC-switching node along the route, the set of available outgoing links (if any) to the destination is determined by checking the available resources, according to the CAC rule in use. If no link can carry it, the connection is dropped and the resources previously allocated along the route are freed. If the set of available links is non-empty, a choice is made among them, by using the mechanism described below.

At the arrival of a connection request (supposed to be at the generic slot k) a generic node i chooses the link to which to forward a class-h connection request, by minimizing (over all successor nodes j; i.e., the subset of neighbouring nodes from which that destination can be reached) the quantity

$$c_{ij}^{(h)}(k,s) = c_{ij,L}^{(h)}(k) + \alpha_j c_j^{(h)}(s) \tag{6}$$

where $c_{ij,L}^{(h)}(k)$ is a local cost related to link ij and $c_j^{(h)}(s)$ is a "global" cost, referring to the traffic conditions of node j and its neighbours at some slot $s < k$. $\alpha_j \in [0, 1]$ is a weighting coefficient used to balance the influence of the local and global cost. $c_{ij,L}^{(h)}(k)$ should weigh the local congestion of link ij, and we have chosen it to be of the following form:

$$c_{ij,L}^{(h)}(k) = \begin{cases} \dfrac{1}{N_{ij,\max}^{(h)}(k) - N_{ij,A}^{(h)}(k)} & \text{if } N_{ij,A}^{(h)}(k) < N_{ij,\max}^{(h)}(k) \\ Z & \text{if } N_{ij,A}^{(h)}(k) = N_{ij,\max}^{(h)}(k) \end{cases} \tag{7}$$

where $N_{ij,\max}^{(h)}(k)$ and $N_{ij,A}^{(h)}(k)$ have been defined in Section 2.4.1, and Z is a very large number; namely, large enough to ensure that no saturated link will be chosen if non-congested links are available. The difference in equation 7 represents, in all these cases, the "available space" on link ij, left to accommodate class-h connection requests.

To complete the description of the algorithm, we define the cost referred to a generic node j in equation 6 to be composed by two terms:

$$c_j^{(h)}(s) = c_{j,L}^{(h)}(s) + \beta_j c_{j,A}^{(h)}(s) \tag{8}$$

where β_j is a weighting coefficient. $c_{j,L}^{(h)}(s)$ represents the average situation of the node with respect to the congestion state of its links, and $c_{j,A}^{(h)}(s)$ is an aggregate information on the average congestion of its successor nodes. More specifically,

$$c_{j,L}^{(h)}(s) = \frac{1}{L_j} \sum_{k \in \mathrm{Out}(j)} c_{jk,L}^{(h)}(s)\ , \tag{9}$$

$$c_{j,A}^{(h)}(s) = \frac{1}{L_j} \sum_{k \in \mathrm{Out}(j)} c_k^{(h)}(s)\ , \tag{10}$$

Out(j) being the set of nodes outgoing from node j. In detail, $c_{j,L}^{(h)}(s)$ represents the average "free space" left for the links outgoing from node j; whereas $c_{j,A}^{(h)}(s)$ is the average of the costs related to each node connected to node j. As can be seen, the values related to the outgoing links nodes are referred to the instants s, where s = T, 2T,..., with T equal to a fixed number of slots. This means that each node i sends its costs $c_i^{(h)}(s)$, $h = 1, \ldots, M$, to its incoming links every T slots and then, after receiving the costs from its outgoing links, recomputes its new aggregate information on the congestion of the network. This function is performed by the simulator by using the event *cost_communication*, with the scheduling time T.

4. Simulation Results

This part is dedicated to showing some examples of possible utilization of the two simulators. In more detail, the first part (Section 4.1) concerns the node simulator, and the second one (Section 4.2) refers to the network simulator.

The data in Table 1 have been used with a channel capacity C = 150 Mbits/s and a related slot duration $T_s = 2.83 \cdot 10^{-6}$ s (53 bytes/cell).

The reference traffic load is $\rho^{(1)} = 80$; $\rho^{(2)} = 100$; $\rho^{(3)} = 40$ [Erlangs]. All the simulations have been stopped with a confidence interval of 3% of the measured values.

4.1 *Node Simulator Test*

Figure 4 shows the real maximum number of acceptable calls obtained by simulations (class 1); the result is compared with the results analytically computed by the access rule in Section 2.4.1 and with the same quantity obtained by allocating each call its peak ($P^{(1)}$) and average bandwidth ($P^{(1)}/b^{(1)}$). The same graph is depicted in Figure 5 concerning traffic class 2. In this case the simulator is used in its simplest form; each traffic class is analyzed one at a time, and the bandwidth for each class is fixed, as is the number of accepted connections (considered of infinite duration here); no *connection_request* occurs. The simulation simply measures, with a determined confidence interval, if, with a certain bandwidth ($V^{(h)}$) and an imposed number of accepted calls ($N_A^{(h)}$), the QoS requirements (P_{loss} and P_{delay}) are satisfied. If the requirements are satisfied, the current number of accepted calls is incremented, otherwise it is decremented, until the maximum number of acceptable calls is determined. These results are very important to test the efficiency of a CAC scheme. For example, it can be noted that the values derived from the access rule are close to the real maximum values for traffic classes 1 and 2.

It is worth mentioning that the results obtained by simulation are used to test the strategy in Section 2.4.1 here, but any other analytical strategy could be used to perform the comparison.

The "feasible region" (in terms of maximum number of acceptable calls for each class) is shown in Figure 6. The figure is obtained by using the admission rule described in Section 2.4.1; in Figure 7 the same quantity obtained by simulation is depicted. In this last case, the values have been obtained as described previously for the individual classes. By comparing the graphs, it can be seen that the values obtained by the access rule are close to those of the simulation and show a satisfactory efficiency for the whole mechanism.

Table 1. Characteristics of the traffic classes used for the tests

TRAFFIC CLASS: h	h=1	h=2	h=3
PEAK BANDWIDTH: $P^{(h)}$	1 Mbits/s	2 Mbits/s	10 Mbits/s
BURSTINESS: $b^{(h)}$	2	5	10
AVERAGE BURST LENGTH: $B^{(h)}$	100 cells	500 cells	1000 cells
AVERAGE CONNECTION DURATION	20 s	15 s	25 s
Ploss UPPER BOUND: $\varepsilon^{(h)}$	0.0001	0.0001	0.0001
Pdelay UPPER BOUND: $\delta^{(h)}$	0.001	0.001	0.001
DELAY CONSTRAINT: $D^{(h)}$	400 slots	200 slots	100 slots
BUFFER LENGTH: $Q^{(h)}$	20 cells	15 cells	10 cells

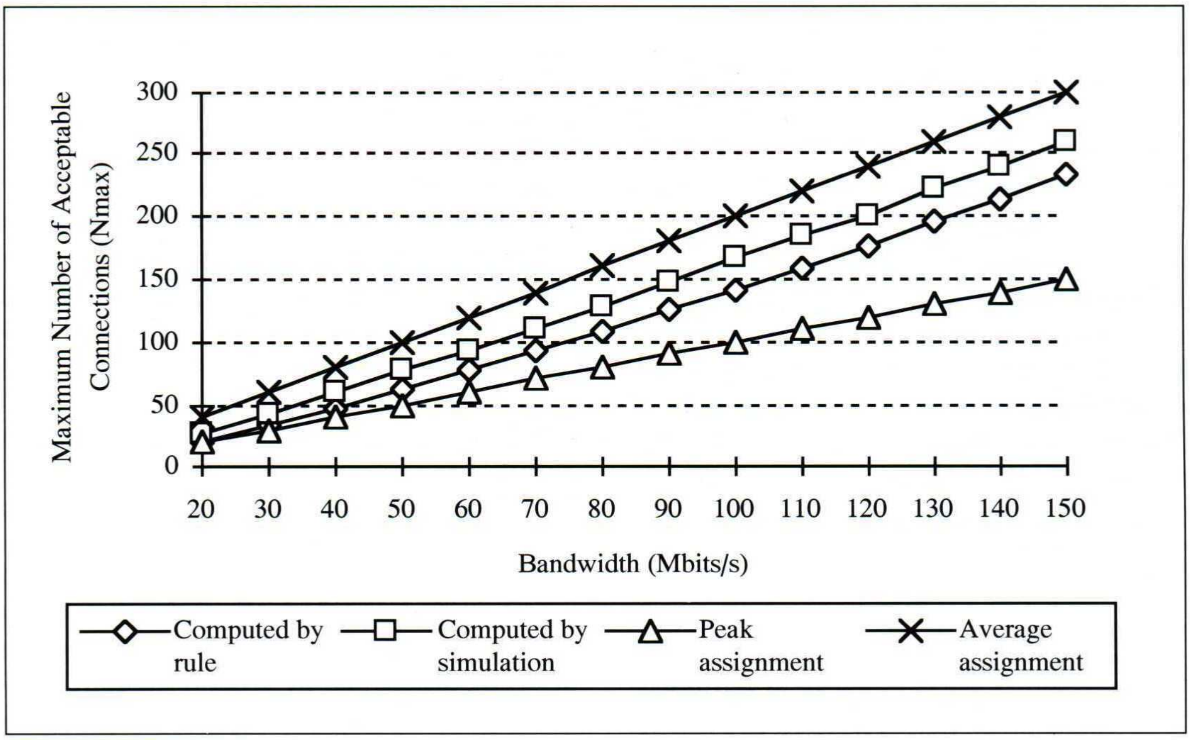

Figure 4. Maximum number of connections versus allocated capacity (class 1)

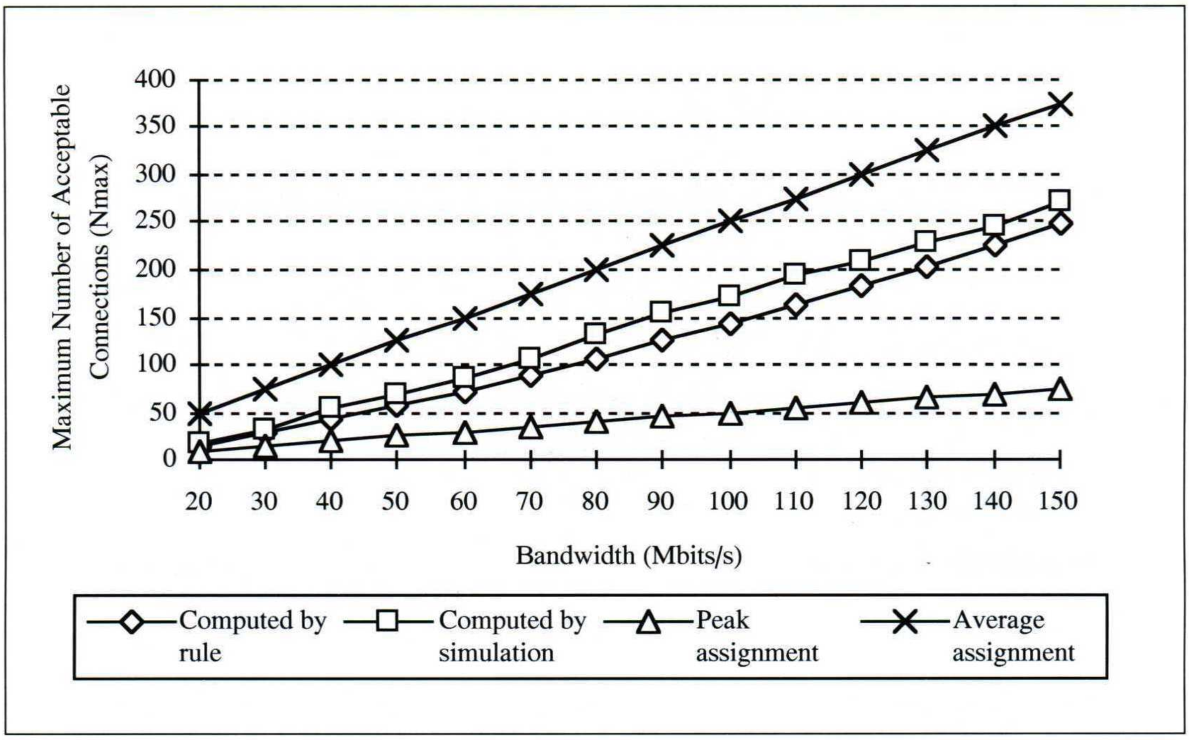

Figure 5. Maximum number of connections versus allocated capacity (class 2)

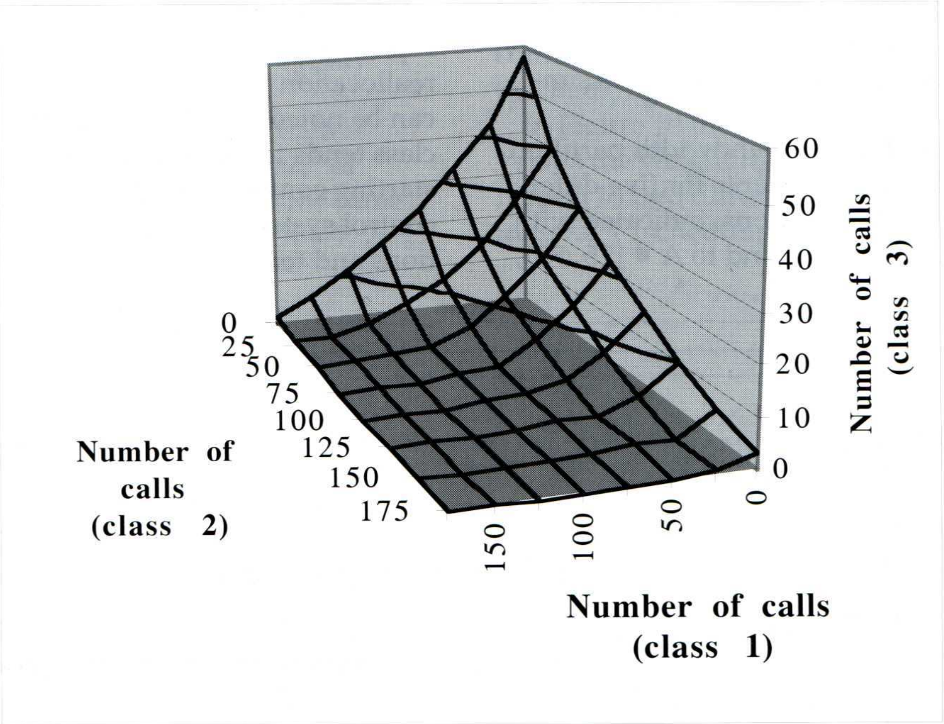

Figure 6. Maximum number of connections by analytical rule

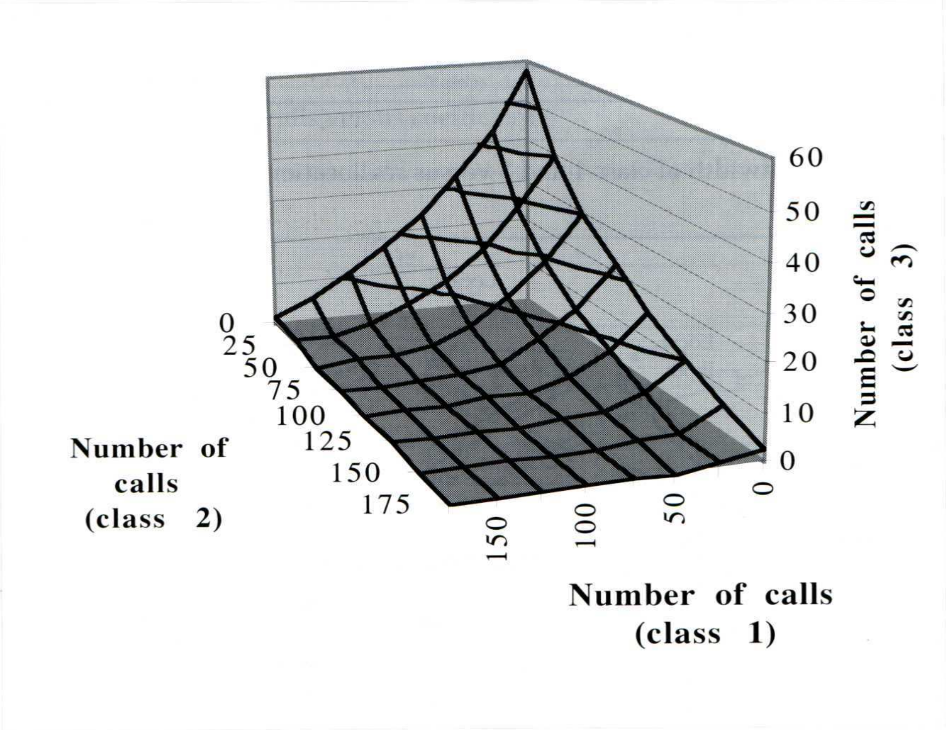

Figure 7. Maximum number of connections by simulation

Another possible application of the simulator is intended to analyze the dynamic behaviour of the allocation mechanism, by using different initial conditions and by changing the offered traffic load dynamically. Now, the simulator is used exactly in the form described in Section 2, and only two classes, namely class 1 and class 3, have been used, to simplify the analysis.

Figure 8 depicts the allocated bandwidth partition values versus the allocations instants for five different initial bandwidth configurations, indicated with A, B, C, D, and E, which correspond to A ≡ [75, 75], B ≡ [100, 50], C ≡ [50, 100], D ≡ [125, 25], and E ≡ [25, 125], respectively; the notation [initial bandwidth class 1, initial bandwidth class 3] has been used. The traffic intensity values for this test are $\rho^{(1)} = 100$ and $\rho^{(3)} = 50$, while the other traffic parameters are the same as reported at the beginning of this section. Each reallocation period K has a duration of $40 \cdot 10^6$ slots. It can be noted that the bandwidth allocated for each class tends to a common value, independently of the starting configuration, and that the whole allocation control system does not depend on the initial conditions and tends to a stable regime.

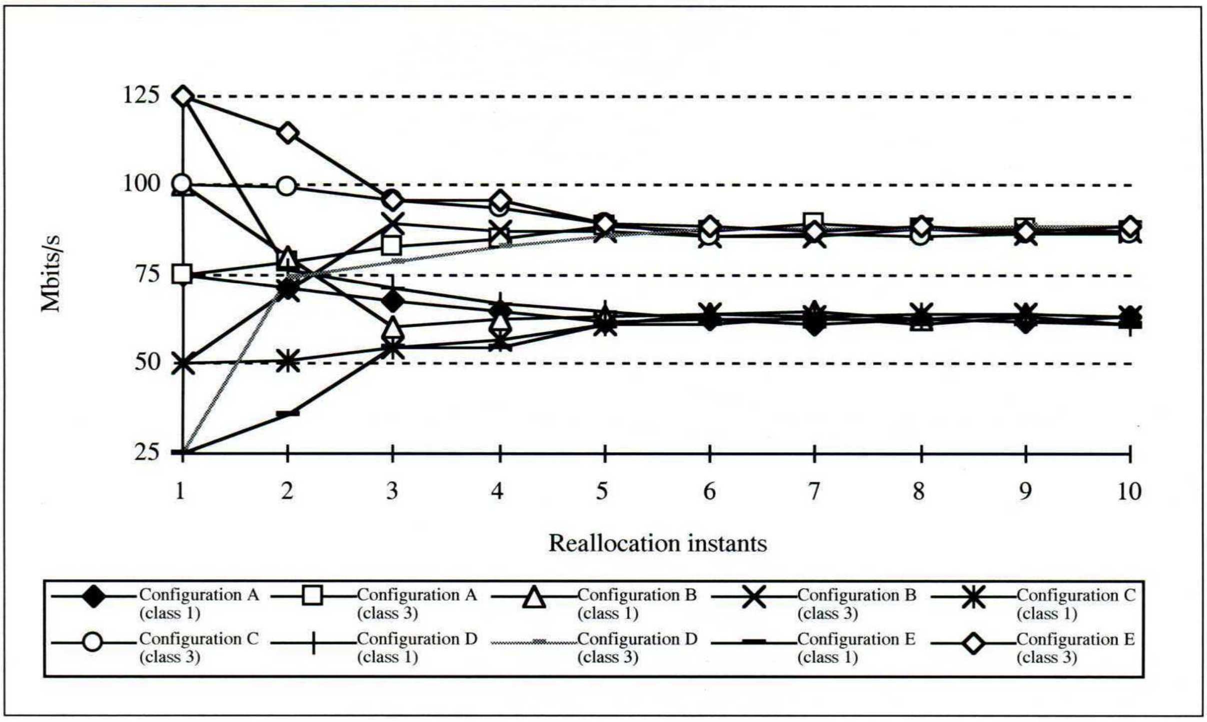

Figure 8. Allocated bandwidth of class 1 and 3 versus reallocation instants with different initial conditions

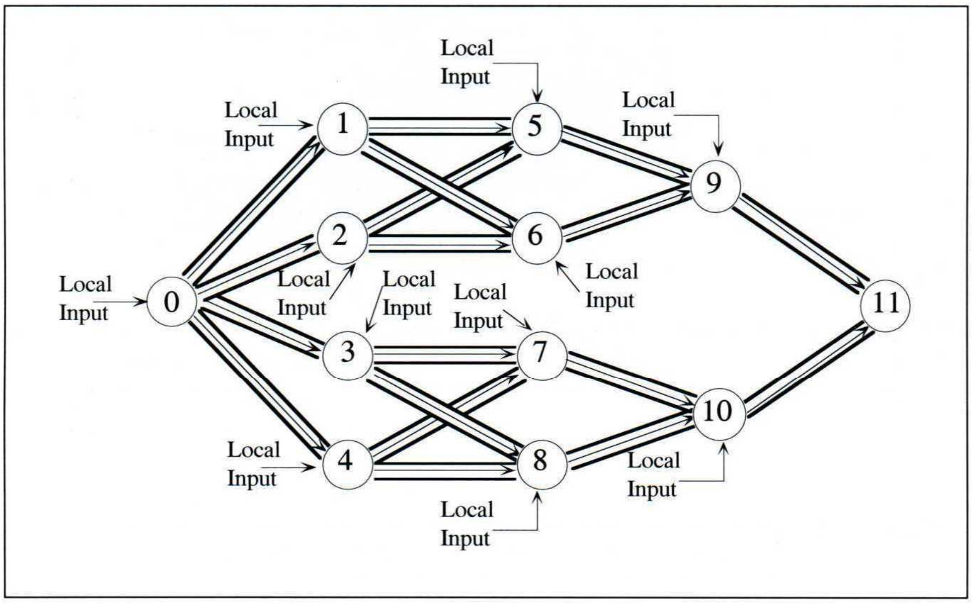

Figure 9. Topology of the first test network

4.2 *Network Simulator Tests*

This part of the section is dedicated to the routing scheme, which is tested and analyzed by using the same data at the beginning of the section (with $\rho^{(1)} = 120$; $\rho^{(2)} = 100$; $\rho^{(3)} = 15$; and $K = 8 \cdot 10^7$ cells) in the twelve-node network depicted in Figure 9 (only node 11 is a destination).

The traffic flow generated by the above data is considered to be a "normalized offered load" of value 1; an offered load, x, corresponds to the same data, except for the traffic intensities $\rho^{(h)}$, h = 1, 2, 3, which are multiplied by x. The coefficients α_i and β_i, i = 0,..., 11, are the same at each node, that is, $\alpha_i = \alpha$ and $\beta_i = \beta$, $\forall$ i. *Resource_allocation* has been scheduled every 226.4 s. Many results can be obtained, both by varying the traffic load or the aggregate cost communication time (*cost_communication* event), and by changing the network topology.

In Figure 10 the percentage of blocked connections versus the weighting coefficient β is shown, with respect to two different values of the updating interval T (scheduling time of *cost_communication*) of

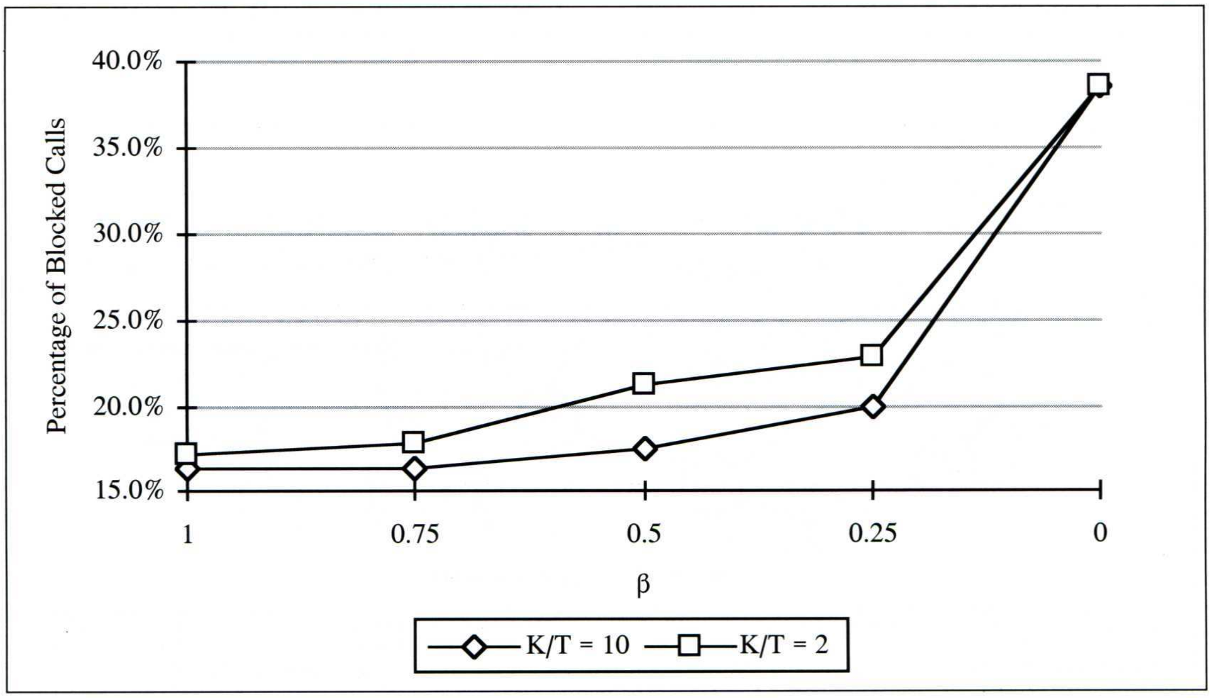

Figure 10. Percentage of blocked calls versus coefficient β

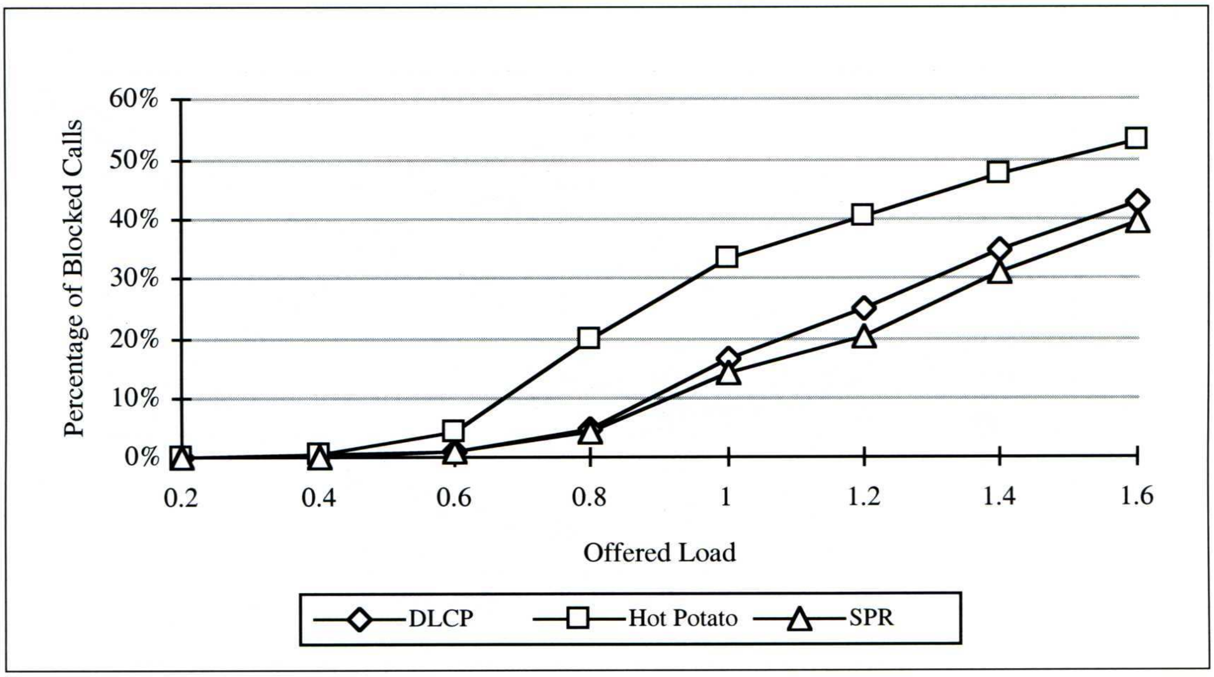

Figure 11. Percentage blocked calls versus offered load

the aggregate information, and a constant value of $\alpha = 1$: the test is performed by updating the aggregate information ten and two times every reallocation interval, respectively ($T = K/10$ and $T = K/2$); it has to be remembered that K is the scheduling time of *resource_allocation*. The improvement obtained by using a more frequent updating is clear, except for $\beta = 0$, where the aggregate information is ignored. In this case, as in the next one, the offered load network configuration is 37.5% to node 0, and 62.5% to node 9, because the aim is to enhance the effect of the knowledge of the aggregate information.

Figure 11 depicts the total percentage of blocked calls versus the offered load. The DLCP routing, with $\alpha = 1$ and $\beta = 1$, is compared with a Shortest Path Routing (SPR) strategy, where the cost of each link is the same as in equation 7, and with a local Hot Potato strategy, which is considered as a possible lower bound on performance. The percentage of blocked calls for DLCP is quite close to that of SPR. The results for SPR and Hot Potato have been obtained by using the same simulation tool, where only the software routine to choose the best path is changed.

A new network topology is shown in Figure 12. The last graph shown (Figure 13) depicts the same quantities (except for the Hot Potato) as in Figure 11, for the topology in Figure 12. In this case all the nodes generate traffic; only traffic classes 1 and 3 (see Table 1) have been used. The traffic reference values are $\rho^{(1)} = 168$, $\rho^{(3)} = 21$ [Erlangs].

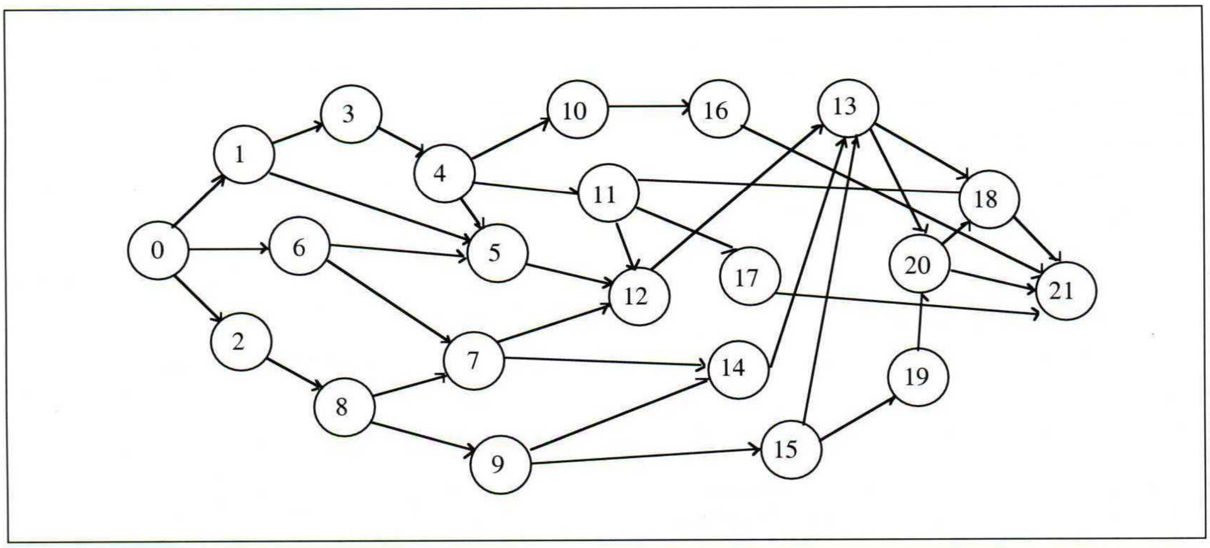

Figure 12. Topology of the second tested network

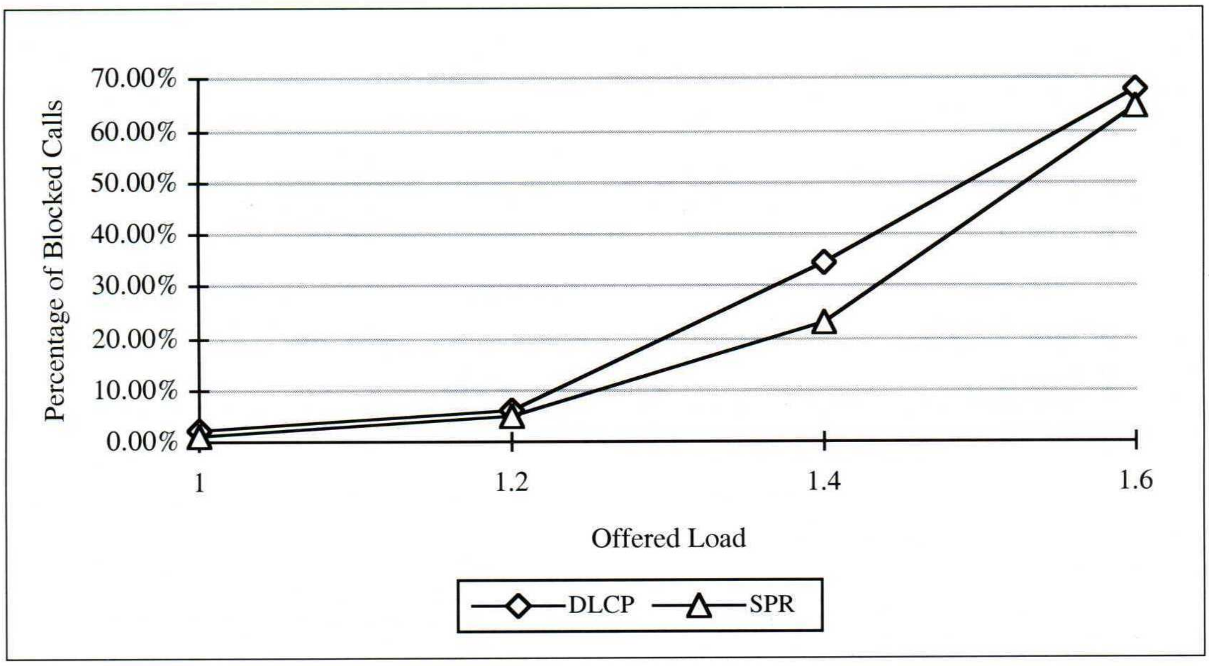

Figure 13. Percentage blocked calls versus offered load

5. Conclusions

Two event-driven simulators have been introduced in the paper. The first one is dedicated to the description of an ATM VC multiplexer, while the second concerns an ATM network. The purpose was to test bandwidth allocation and CAC strategies (in the first case), and routing schemes (in the second case): to this aim, a high level of abstraction was maintained in both cases and many details (e.g., physical transmission and switching fabric) were not considered.

Concerning the node simulator and the traffic generators, a general behavioural description of the ATM node and the events used have been presented. A resource allocation and CAC scheme, utilized in the results, has been summarized. As far as the network simulator is concerned, the model of the network and the events has been explained, along with a specific routing strategy. Simulation results have been presented in both cases, with the purpose of showing some examples of utilization. The node simulator has been used to test and to compare the efficiency of some CAC schemes and to verify the behaviour of a resource allocation scheme. The network simulator has been tested with routing strategies, by changing both some performance parameters and the network topology.

6. References

[1] ITU-T Recommendation G.709. 1993. *Synchronous Multiplexing Structure.* March.

[2] Hui, J.Y. 1990. *Switching and Traffic Theory for Integrated Broadband Networks.* Boston: Kluwer Academic Publishers.

[3] ITU-T Recommendation I.610. 1993. B-ISDN *Operation and Maintenance Principles and Functions.* March.

[4] Hong, D. and Suda, T. 1991. "Congestion control and prevention in ATM networks." *IEEE Network Mag.* vol. 5, no. 4, July, pp. 10–16.

[5] Bolla, R., Davoli, F., Lombardo, A., Palazzo, S., and Panno, D. 1994. "Hierarchical dynamic control of multiple traffic classes in ATM networks." *Europ. Trans. Telecommun.* vol. 5, no. 6, November, pp. 747–755.

[6] Shim, C., Ryoo, I., Lee, J., and Lee, S.B. 1994. "Modelling and call admission control algorithm of variable bit rate video in ATM networks." *IEEE J. Select. Areas Commun.* vol. 12, no. 2, February, pp. 332–344.

[7] Guérin, R., Ahmadi, H., and Naghshineh, M. 1991."Equivalent capacity and its application to bandwidth allocation in high-speed networks." *IEEE J. Select. Areas Commun.* vol. 9, no. 7, September, pp. 968–981.

[8] Bolla, R., Davoli, F., and Marchese, M. 1996. "Simple schemes for traffic integration at call set-up level in ATM networks." *Computer Communications* vol. 19, pp. 645–652.

[9] Bolla, R.,Danovaro, F.,Davoli, F., and Marchese, M. 1993. "An integrated dynamic resource allocation scheme for ATM networks." *Proc. IEEE Infocom '93*, San Francisco, CA, March, vol. 3, pp. 1289–1297.

[10] Kulkarni, V.G., Gün, L., and Chimento, P.F. 1995. "Effective bandwidth vectors for multiclass traffic multiplexed in a partitioned buffer." *IEEE J. Select. Areas Commun.* vol. 13, no. 6, August.

[11] Sidhu, I. and Jordan, S. 1995. "Multiplexing gains in bit stream multiplexors." *IEEE /ACM Trans. Networking* vol. 3, no. 6, December, pp. 785–797.

[12] Wu, G.-L. and Mark, J.W. 1995. "A buffer allocation scheme for ATM networks: Complete sharing based on virtual partitions." *IEEE /ACM Trans. Networking* vol. 3, no. 6, December, pp. 660–670.

[13] Rosberg, Z. 1996. "Cell multiplexing in ATM networks." *IEEE /ACM Trans. Networking* vol. 4, no. 1, February, pp. 112–122.

[14] Chang, C.-S. and Thomas, J.A. 1995. "Effective bandwidth in high-speed digital networks." *IEEE J. Select. Areas Commun.* vol. 13, no. 6, August, pp. 1091–1100.

[15] de Veciana, G., Kesidis, G., and Walrand, J. 1995. "Resource management in wide-area ATM networks using effective bandwidths." *IEEE J. Select. Areas Commun.* vol. 13, no. 6, August, pp. 1081–1090.

[16] Tse, D.N.C., Gallager, R.G., and Tsitsiklis, J.N. 1995. "Statistical multiplexing of multiple time-scale Markov streams." *IEEE J. Select. Areas Commun.* vol. 13, no. 6, August, pp. 1028–1038.

[17] Elwalid, A.I., Mitra, D., and Wentworth, R.H. 1995. "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node." *IEEE J. Select. Areas Commun.* vol. 13, no. 6, August, pp. 1115–1127.

[18] Choudhury, G.L., Lucantoni, D.M., and Whitt, W. 1996. "Squeezing the most out of ATM." *IEEE Trans. Commun.* vol. 44, no. 2, February, pp. 203–217.

[19] Bolla, R., Davoli, F., and Marchese, M. 1995. "A distributed strategy for CAC and routing in ATM networks." *Proc. First IFIP Workshop on ATM Traffic Management* (*WATM'95*), Paris, France, December, pp. 357–363.

[20] Gupta, S., Ross, K.W., and El Zarki, M. 1992. "Routing in virtual path based ATM networks." *Proc. IEEE Globecom'92*, Orlando, FL, December, pp. 571–575.

[21] Gupta, S. and Gandhi, P.P. 1994. "Dynamic routing in multi-class non-hierarchical networks." *Proc. IEEE Internat. Conf. Commun.* (*ICC '94*), New Orleans, LA, May, pp. 1390–1394.

[22] Hyman, J.M., Lazar, A.A., and Pacifici, G. 1994. "A methodology for Virtual Path and Virtual Network bandwidth assignment in broadband networks with QOS guarantees." *Proc. IEEE Internat. Conf. Commun.* (*ICC '94*), New Orleans, LA, May, pp. 165–169.

[23] Maunder, A. and Min, P.S. 1994. "Routing for multirate traffic with multiple qualities of service." *Proc. Internat. Conf. on Computer Commun. and Networks* (*ICCCN'94*), San Francisco, CA, September, pp. 104–108.

[24] Chen, W.-T. and Liu, U.-J. 1994. "Routing problem with performance requirement translation for multimedia communications in an ATM wide-area network." *Proc. IEEE Internat. Conf. Commun.* (*ICC '94*), New Orleans, LA, May, pp. 1490–1493.

[25] Plotkin, S. 1995. "Competitive routing of virtual circuits in ATM networks." *IEEE J. Select. Areas Commun.* vol. 13, August, pp. 1128–1136.

[26] Murakami, K. and Kim, H.S. 1996. "Virtual Path routing for survivable ATM networks." *IEEE/ACM Trans. Networking* vol. 4, February, pp. 22–39.

[27] Alqaed, A.M. and Chang, C.H. 1994. "Simulation driven resource allocation in ATM networks." *Simulation Digest* vol. 24, no. 2, pp. 2–19.

[28] Freebersyser, J.A. and Townsend, J.K. 1996. "Efficient simulation of cell loss probability in ATM networks with heterogeneous connection traffic descriptor." *Proc. IEEE Internat. Conf. Commun.* (*ICC'96*), Dallas, TX, June , pp. 320–327.

[29] Aly, K.A. and Dowd, P.W. 1991. "Simulation of fast packet-switched photonic networks for interprocessor communication." *24th Annual Simulation Symp.*, New Orleans, LA, April, pp. 301–311.

[30] Lin, Y.-B. 1994. "Parallel independent replicated simulation on a network of workstations." *Proc. 8th Workshop on Parallel and Distributed Simulation* (*PADS'94*), Edinburgh, Scotland, U.K., July, pp. 73–80.

[31] Huang, C., Devetsikiotis, M., Lambadaris, I., and Kaye, A.R. 1995. "Fast simulation for self-similar traffic in ATM networks." *Proc. IEEE Internat. Conf. Commun.* (*ICC'95*), Seattle, WA, June, pp. 438–444.

[32] Kumar, A. and Singh, S. 1992. "Simulation based performance study of a dynamic routing algorithm." *Simulation Digest* vol. 22, no. 1, pp. 28–38.

[33] Pitts, J.M. and Schormans, J.A. 1996. *Introduction to ATM Design and Performance.* Chichester: John Wiley & Sons.

[34] Garzia, M.R. 1990. "Discrete event simulation methodologies and formalisms." *Simulation Digest* vol. 21, no. 1, pp. 3–13.

[35] Pollacia, L.F. 1989. "A survey of discrete event simulation and state-of-the-art discrete event languages." *Simulation Digest* vol. 20, no. 3, pp. 8–25.

[36] Fishman, G.S. 1978. *Principles of Discrete Event Simulation.* New York: John Wiley & Sons.

[37] Banks, J. and Carson, J.S. 1984. *Discrete-Event System Simulation.* Englewood Cliffs, NJ: Prentice-Hall.

[38] Kleijnen, J.P.C. 1987. *Statistical Tools for Simulation Practitioners.* New York: Marcel Dekker.

[39] Ross, K.W. 1995. *Multiservice Loss Models for Broadband Telecommunication Networks.* London: Springer-Verlag.

[40] Cosmas, J.P., et al. 1994. "A review of voice, data and video traffic models for ATM." *Europ. Trans. Telecommun.* vol. 5, no. 2, March-April, pp. 11–26.

[41] Yang, T. and Tsang, H.K. 1995. "A novel approach to estimating the cell loss probability in an ATM multiplexer loaded with homogeneous on-off sources." *IEEE Trans. Commun.* vol. 43, no. 1, January, pp. 117–126.

[42] Bolla, R., Davoli, F., and Marchese, M. 1996. "Evaluation of a cell loss rate computation method in ATM multiplexers with multiple bursty sources and different traffic classes." *Proc. IEEE Globecom'96*, London, U.K., November, pp. 437–441.

RAFFAELE BOLLA was born in Savona, Italy, in 1963. He received a "laurea" degree in Electronic Engineering from the University of Genoa in 1989 and a PhD degree in Telecommunications at the Department of Communications, Computer and Systems Science (DIST) of the University of Genoa, in 1994. From 1994 to 1996 he was a postdoctoral research fellow at DIST. Since November 1996 he has been an Assistant Professor in the same department. His current research interests are in management and control of STM and ATM networks, multimedia communications, and multiple access in integrated mobile radio networks. He is a member of IEEE.

AHMAD DALAL'AH was born in Beit Yafa, Jordan, in 1957. He got his BSc in Computer Science from Yarmouk University, Jordan, in 1985. From 1986 to 1988 he worked as a high school teacher in Irbid, Jordan; from 1989 to 1993 he was a teaching assistant at the Computer Department of Mu'tah University, Jordan; since 1994 he has been studying for his PhD degree at the Department of Communications, Computer and Systems Science (DIST), University of Genoa, Italy. He is an IEEE member.

FRANCO DAVOLI was born in Genoa, Italy, in 1949. He received a "laurea" degree in Electronic Engineering from the University of Genoa in 1975. Since 1985 he has been an Associate Professor and, since 1990, a Full Professor of Telecommunication Networks at the University of Genoa, where he is with the Department of Communications, Computer and Systems Science (DIST). From 1989 to 1991 and from 1994 to 1996, he was also with the University of Parma, Italy, where he taught a class in Telecommunication Networks, by means of interactive distance learning over ISDN. His current research interests are in bandwidth allocation, admission control and routing in STM and ATM networks, multimedia communications and services, and integrated services mobile radio networks. He is a member of IEEE.

MARIO MARCHESE was born in Genoa, Italy, in 1967. He received a "laurea" degree in Electronic Engineering from the University of Genoa, Genoa, Italy, in 1992. He has concluded his PhD in Telecommunications at the Department of Communications, Computer System Science (DIST), University of Genoa. His research interests include traffic modelling, and admission control and routing in ATM-based networks. He is a member of IEEE.