Simple Schemes for Traffic Integration at Call Set-Up Level in ATM Networks

R. Bolla, F. Davoli, M. Marchese

Department of Communications, Computer and Systems Science (DIST)
University of Genoa
Via Opera Pia, 13 - 16145 Genova, Italy

Abstract

The paper presents a scheme for admission control at the call set-up in an ATM node, aimed at minimising the probability of blocking a call and at balancing this probability between traffic classes. The traffic is considered to be divided into traffic classes, characterised by statistical parameters like peak and average bandwidth and by Quality of Service (QoS) requirements that allow to define a 'feasibility region', where requirements are guaranteed. A model to describe the 'call admission' is proposed; two different cost functions, based on the blocking probability, are defined and minimised, taking into account the 'feasibility region' constraint: the first function is intended to obtain the minimum overall blocking probability, whereas the second is aimed to balance the blocking probability among classes. The maximum number of acceptable calls yielded by each cost function is obtained, for each class, by minimising it under stationary traffic conditions.

The efficiency of the proposed strategy is tested by simulations and verified by comparing it with other admission schemes.

1. Introduction

In an ATM environment, broadband integrated services networks will carry bursty traffic with different characteristics and Quality of Service (QoS) requirements. Many studies and experiments are currently performed, and many topics investigated in order to satisfy requirements for any application. Fair resource allocation strategies and optimised bandwidth management are a subject widely treated in the literature [1-4]. In [5-10] Call Admission Control (CAC) strategies are investigated and tested, while in [11-12] scheduling algorithms providing satisfactory QoS to each service are proposed. Routing strategies are proposed in [13, 14] to obtain a fair resource management. The problems mentioned above are not independent, and, even if each of them is treated separately from the others, they can be seen in a unified way as in [15-22]. Other considered problems are the buffer position (input, output or input-output queueing) in an ATM node (discussed in [23]) and the segregation of the traffic in classes characterised by different parameters as in [11,17,19-22,24] among the others.

In this paper an output queueing node model is used; the traffic is divided into traffic classes defined by peak and average bandwidth and by QoS requirements (as cell loss and delayed cell rate [19-22], for example). The CAC mechanism is structured by considering a maximum number of acceptable connections that allow to guarantee performance requirements to each traffic class, retaining the general philosophy proposed in [19-22].

The paper is structured as follows: the next Section is dedicated to describe the system model and the CAC scheme; in Section 3 cost functions, aimed to minimise and to balance the call blocking probability, are defined, while Section 4 contains simulation results and some comparisons with other CAC strategies. Conclusions are provided in Section 5.

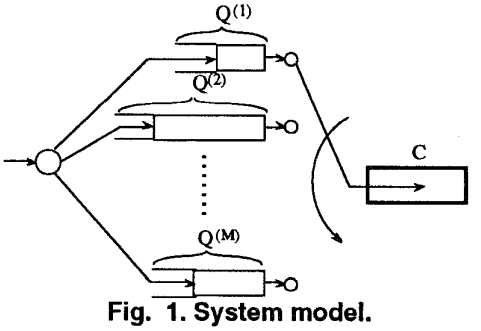

Fig. 1. System model.

2. System model and CAC scheme

The node model is supposed to be an output queuing one, and, for simplicity, only one output link is considered (see [21, 22], for an extension). The system model is shown in Fig. 1, where the traffic entering the node is queued in buffers, one for each traffic class, with

0-8186-7180-7/95 $04.00

limited length. The buffer length is fixed to the value $Q^{(h)}$, h=1..M, and the capacity of the output link is assumed to be C.

In correspondence of each output link a 'Call Admission Control Block' decides if a connection has to be accepted or rejected, while a 'Scheduler' picks up the cells of the output buffer following a strategy presented in [19, 20] which essentially grants each traffic class a fixed portion of the total bandwidth. The CAC follows a simple strategy explained in the following.

Traffic entering the node is considered segregated into traffic classes, characterised by statistical parameters (for class (h), peak bandwidth - $B_p^{(h)}$, average bandwidth - $B_m^{(h)}$ and, as a consequence, burstiness - $b^{(h)}=\frac{B_p^{(h)}}{B_m^{(h)}}$) and by performance requirements that allow to define a 'feasibility region', a region in call space, where performance requirements are statistically guaranteed. The 'feasibility region' can be obtained by complex analytical models or by simulations, and many studies can be found in the literature, among many others [1,11,17,25] and [26, 27] where a survey of classes of control policies and a characterisation of the optimal strategy in the specific case of 'Coordinate Convex Policies' is also reported. The region depicted in Fig. 3 has been computed by using the strategy extensively presented in [19, 20], which can be summarised as follows: an ON/OFF model is used for each bursty source and the probability of generating a cell in the active state (ON) for a call of class h is $\frac{B_p^{(h)}}{C}$. Each traffic class (h) has to be guaranteed with two performance requirements, the cell loss ($P_{loss}^{(h)}(n^{(h)})$) and the delayed cell rate ($P_{delay}^{(h)}(n^{(h)})$), supposing to have $n^{(h)}$ calls in the active state. Being $v_{n^{(h)},N^{(h)}}$ the probability of having $n^{(h)}$ connections in the active state out of $N^{(h)}$ accepted connections, performance requirements can be considered to be:

$$\sum_{n^{(h)}=1}^{N^{(h)}} P_{loss}^{(h)}(n^{(h)})\, v_{n^{(h)},N^{(h)}} \le \varepsilon^{(h)} \qquad (1)$$

$$\sum_{n^{(h)}=1}^{N^{(h)}} P_{delay}^{(h)}(n^{(h)})\, v_{n^{(h)},N^{(h)}} \le \delta^{(h)} \qquad (2)$$

where $\varepsilon^{(h)}$ is an upper limit on the time-averaged value of the cell loss rate and $\delta^{(h)}$ has the same meaning for the time-averaged value of cells that suffer a delay longer than a fixed value ($D^{(h)}$ in Section 4).

The two inequalities above,(1) and (2), define a region in call space where performance requirements are satisfied, that is, they determine a 'feasibility region'. It should be pointed out that the chosen region is just an example and the strategy to individuate the 'feasibility region' does not affect the overall call acceptance scheme defined in the following and in the next Section.

The admission strategy can be summarised as follows: a maximum number of acceptable connections $N_{max}^{(h)}$, whose computation is presented in the next Section, shall be defined for each class. $N_c^{(h)}$ being the number of accepted connections, a new call is accepted if the number of calls in progress in the node and the new one does not exceed the maximum number of acceptable calls for that traffic class. It can be said that:

$$\begin{aligned} N_c^{(h)} + 1 \le N_{max}^{(h)} \quad & \text{accepted} \\ N_c^{(h)} + 1 > N_{max}^{(h)} \quad & \text{rejected} \end{aligned} \qquad (3)$$

So, the CAC blocking can be modelled, for the generic class (h), as in Fig. 2

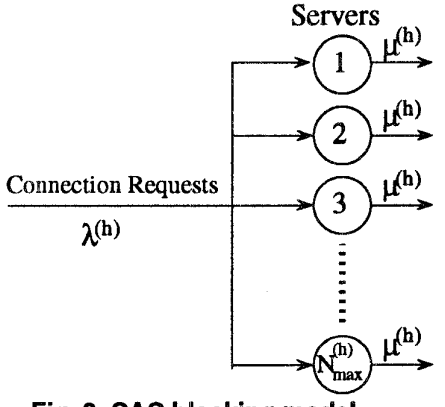

Fig. 2. CAC blocking model.

where the number of servers is equal to the maximum number of acceptable calls of that class. The quantity $\lambda^{(h)}$ represents the average call arrival rate, according to a Poisson process, while $\frac{1}{\mu^{(h)}}$ is the average duration of a call, which is supposed to be exponentially distributed. The quantity $N_a^{(h)} = \rho^{(h)} = \frac{\lambda^{(h)}}{\mu^{(h)}}$ (in Erlangs) is the average traffic intensity for traffic class (h).

Considering the number $N_{max}^{(h)}$ fixed the probability of blocking a call is given by the well known Erlang-B distribution:

$$P_B^{(h)} = \frac{(N_a^{(h)})^{N_{max}^{(h)}} / N_{max}^{(h)}!}{\sum_{l=0}^{N_{max}^{(h)}} \frac{(N_a^{(h)})^l}{l!}} \quad (4)$$

This probability is used in the next Section, where cost functions are defined and the strategy to find the maximum number of acceptable connections is presented.

3. Cost function definitions

In this paper, two different cost functions are proposed with two specific purposes (a detailed discussion about different possible choices in a similar context is contained in [28]). The first one is intended to minimise the overall call blocking probability, so, the definition of the cost function is:

$$P_B = \sum_{h=1}^{M} \alpha^{(h)} P_B^{(h)} \quad (5)$$

being M the number of traffic classes, $P_B^{(h)}$ the blocking probability of class h, defined in the previous Section, and $\alpha^{(h)}$ a weight to allow a distinct priority level to each traffic class. However, in the simulation tests performed in Section 4, $\alpha^{(h)}=1$ for each traffic class (h).

By the minimisation of P_B over the 'feasibility region', the maximum number of acceptable connections for each traffic class has to be obtained. So, defining the 'feasibility region', as the set F_R of M-tuple $\overline{N} \equiv col[N^{(h)}, h=1..M]$ that satisfy performance requirements ((1) and (2), in this context) it can be said that:

$$\overline{N}_{max} \equiv col\left[N_{max}^{(h)}, h = 1..M\right] = \arg\min_{\overline{N} \subset F_R} P_B \quad (6)$$

It should be mentioned that this function, as it will be clarified in the next Section, has the only aim of minimising the overall call blocking probability without balancing in any way the blocking among traffic classes. In the following, this CAC scheme is named Erlang Scheme (ES).

If the balancing among classes is the purpose, another cost function has to be used. A possible choice is:

$$P'_B = \max_h P_B^{(h)} \quad (7)$$

$N_{max}^{(h)}$ can be obtained in the same way as in (6), by substituting the quantity P'_B to the quantity P_B. In formula:

$$\overline{N}_{max} \equiv col\left[N_{max}^{(h)}, h = 1..M\right] = \arg\min_{\overline{N} \subset F_R} P'_B \quad (8)$$

The scheme using (8) is called Balanced Erlang Scheme (BES), in the following. In the next Section, simulation results can help to appreciate the difference between the two presented cost functions.

It has to be remembered that, in this paper, the values $\lambda^{(h)}$ and $\mu^{(h)}$, defined in the previous Section, are considered to remain constant, so that the minimisation procedure is performed just once. If the characteristic of the traffic should vary, the minimisation should be performed again.

4. Results

The purpose of this Section is to investigate the difference between the two proposed cost functions and to test and verify the efficiency of the overall strategy by showing simulation results. Four traffic classes with quite different characteristics have been chosen to obtain the results in the following. The first traffic class represents medium quality video and the second bulk data transfer while the third and the fourth class denote, respectively, voice and image retrieval as in [29]. For the sake of simplicity each test has been carried out with a couple of traffic classes and only the most meaningful results have been depicted. The following traffic classes have been used:

$B_p^{(1)}$=1 Mbit/s; $B_p^{(2)}$=10 Mbit/s;
$B_p^{(3)}$=64 Kbit/s; $B_p^{(4)}$ =2 Mbit/s; (peak bandwidth)
$b^{(1)} = 2$; $b^{(2)} = 10$; $b^{(3)} = 2$; $b^{(4)} = 23$ (burstiness)
$B^{(1)}$=100; $B^{(2)}$=1000; $B^{(3)}$=58; $B^{(4)}$=2604 cells (average burst length)
$1/\mu^{(1)} = 20$ s; $1/\mu^{(2)} = 25$ s; $1/\mu^{(3)} = 30$ s; $1/\mu^{(4)} = 60$ s (average connection duration)
$\varepsilon^{(1)} = \varepsilon^{(2)} = 1 \cdot 10^{-4}$; $\varepsilon^{(3)} = \varepsilon^{(4)} = 1 \cdot 10^{-6}$ (upper limit for the average cell loss rate)
$\delta^{(1)} = \delta^{(2)} = 1 \cdot 10^{-3}$; $\delta^{(3)} = \delta^{(4)} = 1 \cdot 10^{-5}$ (upper limit for the average delayed loss rate)
$D^{(1)}$=400; $D^{(2)}$=100 slots; $D^{(3)}$=200; $D^{(4)}$=1000 slots (delay threshold)
$N_a^{(1)}$=80; $N_a^{(2)}$=40; $N_a^{(3)}$=250; $N_a^{(4)}$=92 Erlangs (global average traffic intensities offered to the network; call arrival processes follow independent Poisson distributions)
$Q^{(1)} = 20$; $Q^{(2)} = 10$; $Q^{(3)} = 15$; $Q^{(4)} = 15$ cells (buffer length)

Figs. 3, 5, 7, 8 are referred to the couple class 1- class 2 (called couple A in the following); a channel capacity C = 150 Mbit/s (T_S = slot duration = $2.83 \cdot 10^{-6}$ s (53 bytes/cell)) has been used and the duration of the performed simulations corresponds to 37 minutes and 44 s of real network time. Figs. 4, 6, 9, are obtained by using the couple class 3- class 4 (called couple B in the following), a channel capacity C = 30 Mbit/s (T_S = slot duration = $14.1 \cdot 10^{-6}$ s (53 bytes/cell)). The duration of the performed simulations corresponds, in this case, to 1 hour, 14 minutes and 13 s of real network time. The simulations have been carried out on a Sun Sparc 10 workstation.

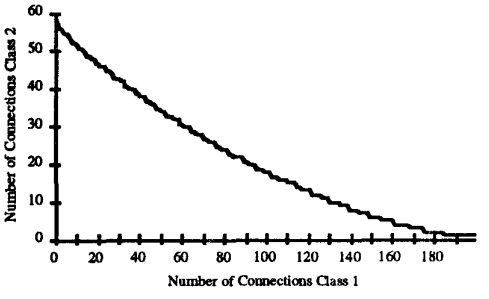

Fig. 3. Feasibility region

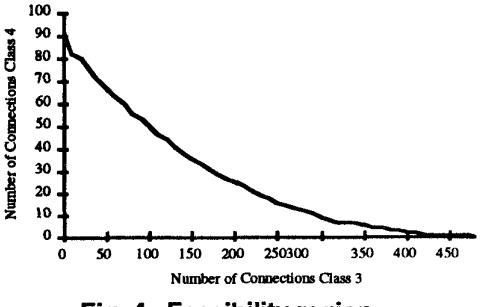

Fig. 4. Feasibility region

Figs. 3 and 4 depict the 'feasibility region' for the couple A and B, respectively. The regions have been obtained by using the inequalities (1) and (2) and the data reported above; however any other 'feasibility region', obtained by using different techniques, could be utilised without affecting the global strategy.

The traffic flow generated by the above data is considered to be a 'normalised offered load' of value 1; an offered load 'x' corresponds to the same data, except for the traffic intensities (defined in Section 2) $N_a^{(h)}$, h=1, 2, 3,4 ,which are multiplied by x.

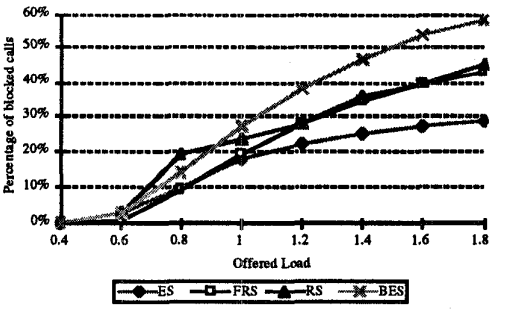

Fig. 5 . Overall percentage of blocked calls versus offered load

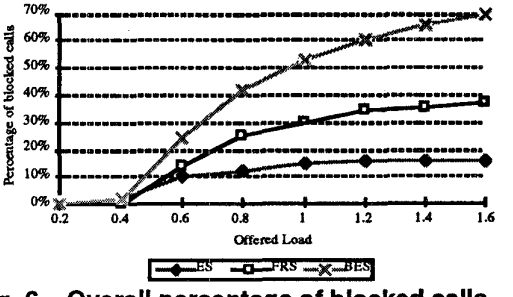

Fig. 6 . Overall percentage of blocked calls versus offered load

The overall percentage of blocked calls versus the offered load is depicted in Figs. 5 (couple A) and 6 (couple B). Results obtained by using ES and BES are compared with the results obtained by using a strategy, called Feasibility Region Scheme (FRS) here, that does not compute a maximum number of acceptable connections, but always accepts a call with the only constraint of the 'feasibility region'; that is, if a new call in the system keeps the total number of connections within the 'feasibility region', the call is accepted. In Fig. 5 is also reported a comparison with the strategy reported in [19], called Reallocation Scheme (RS) here. The improvement in the percentage of blocked calls by using ES is noticeable; in fact, the cost function P_B has the only purpose of minimising the overall call blocking probability without consideration of balancing among classes.

This behaviour can be better explained by observing Fig. 7 (couple A), where the percentage of blocked calls for each traffic class is shown versus the offered load, by using the ES, FRS and RS. It can be seen that the blocking percentages of the two classes are completely unbalanced, particularly for the ES case. So the low overall call blocking percentage is 'paid' with a conspicuous unbalanced effect.

On the contrary, the utilisation of BES does not guarantee the lowest overall percentage of blocked calls, but it assures a fair division among the classes, as shown in Figs. 8 and 9. These figures depict the same

quantities of the previous figure versus the offered load, for couple A (Fig. 8), by using BES, FRS and RS, and for couple B (Fig. 9), by using BES, ES and FRS.

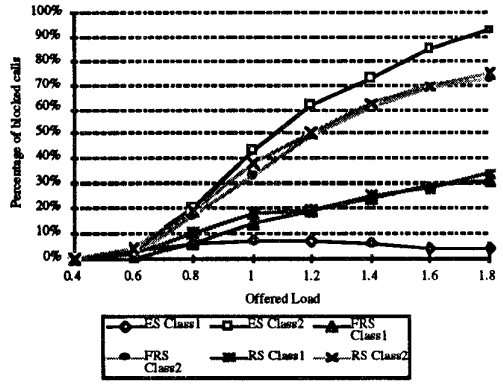

Fig. 7. Percentage of blocked calls for each traffic class versus the offered load by using the ES, FRS and RS.

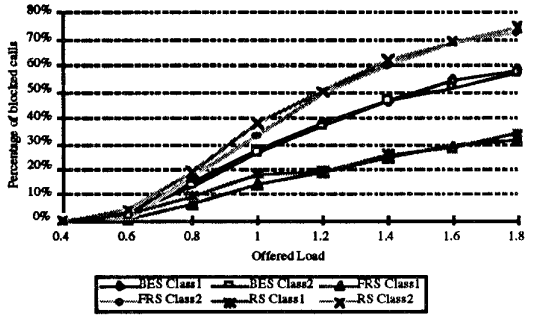

Fig. 8. Percentage of blocked calls for each traffic class versus the offered load by using the BES, FRS and RS.

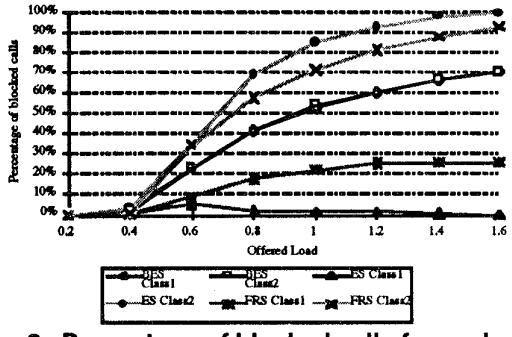

Fig. 9. Percentage of blocked calls for each traffic class versus the offered load by using the ES, FRS and BES.

The choice between the two cost function has to depend on the application and on user requirements.

The problem of minimising the blocking probability without an excessive unbalanced effect could be solved by imposing some constraints to the call blocking probability and evaluating a sub-region inside the feasibility region where the constraints are satisfied. However, this topic is currently investigated and it will be object of future work.

5. Conclusions

In this paper two simple CAC schemes for traffic integration in ATM networks at the call set-up level are proposed. The first scheme, named Erlang Scheme (ES), is aimed to minimise the overall call blocking probability, while the second one, named Balanced Erlang Scheme (BES), has the main purpose of balancing the number of blocked calls among traffic classes. In both schemes the minimisation is performed by taking into account the constraint of the 'feasibility region'.

Four traffic classes, with different characteristics, are considered to test the proposed strategies. Simulation results have shown the difference of the two proposed schemes, and verified the good efficiency of ES and BES with respect to other CAC strategies already presented in the literature. More specifically ES allows the lowest overall call blocking probability, while BES offers a very good balancing among the different traffic classes. The extreme simplicity of the cost functions makes the computation quite fast and the algorithms well suited for control mechanisms.

References

[1] R.Guérin, H.Ahmadi, M.Naghshineh, 'Equivalent capacity and its application to bandwidth allocation in high speed networks', *IEEE J. Select. Areas Commun.*, vol. 9, n. 7, pp. 968-981, Sept. 1991.

[2] F.P.Kelly, 'Effective bandwidths at multi-class queues', *Queueing Systems*, vol. 9, pp. 5-15, 1991.

[3] R.J.Gibbens, P.J.Hunt, 'Effective bandwidths for the multi-type UAS channel', *Queueing Systems*, vol. 9, pp. 17-27, 1991.

[4] A.Suruagy Monteiro, M.Gerla, 'Bandwidth allocation in ATM networks', *Annals of Operations Research*, vol 49, pp. 25-50, 1994.

[5] H.Saito, Teletraffic Technologies in ATM Networks, Artech House, Inc., Norwood, MA, 1994.

[6] R.O.Onvural, Asynchronous Transfer Mode Networks: Performance Issues, Artech House, Inc., Norwood, MA, 1994.

[7] T. Kamitake, T. Suda, 'Evaluation of an admission control scheme for an ATM network considering