# Simulation Study of CAC Algorithms for ATM Networks

Niudomar Siqueira de A. Chaves, Shusaburo Motoyama

Department of Telematics, School of Electrical and Computer Engineering,
University of Campinas – UNICAMP, P.O. Box 6101, 13083-970, Campinas, SP, Brazil,
Phone +55 19 3788 3765, Fax +55 19 3289 1395.
E-mails: {niudomar, motoyama}@dt.fee.unicamp.br.

*Abstract* - **In this paper a performance study of three CAC (Connection Admission Control) algorithms for ATM networks is presented. The results are obtained through discrete event simulation. Two of the algorithms are based on effective bandwidth concept and the other is a measurement-based method, which uses the actual traffic besides the formal parameters. The analysis showed that all the algorithms achieve the required QoS, however they overestimate the necessary bandwidth in several situations, resulting in lower network resource utilization.**

## I. INTRODUCTION

MANY different network solutions have been proposed to integrate several services into only one network denoted integrated services network. The main objective of the integrated services network is the transport of all kind of services (such as voice, data, video, sound, etc.), and it also must guarantee the QoS (quality of service) of each service. Most important network solutions for integrated services network are ATM (asynchronous transfer mode), DiffServ (differentiated services) and MPLS (MultiProtocol Label Switching).

ATM has received much attention since 80´s and was adopted by ITU-T, as the transport network for B-ISDN (broadband integrated services digital network). Many researches have been carried out to implement all aspects of the ATM network and now ATM is widely adopted in backbone applications. In ATM the services are divided into classes of service to facilitate the assurance of QoS of each service.

DiffServ network was proposed to guarantee the QoS of services for IP network, which is based on best effort delivery of packets. The ingress node in DiffServ network classifies the packets according to a small list of classes of services, which aggregates many different flows in a small set of levels of QoS. Each core node routes the packets based on their classes of service.

MPLS based network combines the high-speed layer 2 switching technique with the layer 3 IP routing technique. In MPLS each packet receives a label for layer 2 switching, as the cell header in ATM, and each packet is classified into forwarding equivalence class (FEC). In each network node a path is defined for each FEC, using a routing protocol (layer 3), such as OSPF (open shortest path first). The packet classification and the establishments of paths facilitate the network scalability and the assurance of QoS.

In all three networks above a way to guarantee the QoS is to reserve the network resources based on classes of service, or flows, or forwarding equivalence classes. Since the cells or packets are generated by terminals in random bases, the optimum quantity of resource for each class of service, or for each flow, or for each FEC is difficult to estimate. But, when this estimation is made, that resource is kept for a while. Thus, a situation may occur in which a quantity of resource needed for a new path to be established is not available. In this case, that path is not established, and some kind of loss will occur (for instance, a new terminal is not allowed to be connected). This network function that monitors the quantity of available resources and evaluates the needs for a new connection to satisfy the required QoS is named connection admission control (CAC).

In this paper, CAC for ATM networks is analyzed. Several methods have been proposed for the implementation of CAC in ATM networks. Most of them are based on effective bandwidth concept [1, 2], which means the necessary bandwidth for one connection alone.

Other methods use dynamic CAC schemes [3, 4, 5]. The decision about the acceptance is based on real time measurement of traffic. Le Boudec and Nagarajan [6] proposed a deterministic method that considers the worst case cell transfer delay.

The objective of this work is to evaluate through discrete event simulation the algorithms based on effective bandwidth proposed by Guérin, Ahmadi and Naghshineh [1] (referred to as GAN throughout this paper), the one proposed by Kesidis, Walrand and Chang (KWC) [2] and the dynamic method proposed by Saito and Shiomoto (SS) [3].

In section II, the three algorithms for CAC are presented. The simulation objectives and the results of simulation are discussed in section III. Finally, the main conclusions are presented in section IV.

## II. ALGORITHMS

### *A. GAN Algorithm*

The scheme proposed by Guérin, Ahmadi and Naghshineh [1] uses the on-off traffic model. The on and off duration states have geometric distribution. During on state, there is cell generation at a constant rate and the off state is a silent period.

An on-off source can be characterized by its peak rate $R$, its load $r$ and its mean burst length $b$. The switch model used by [1] is based, initially, on a single on-off source and a finite capacity queue (buffer) with constant service time, shown in Fig. 1.

Let $P_i(t,x)$ be the probability of the source be in state $i$ when the occupancy of the buffer is $x$ at the instant $t$. If $i = 1 \Rightarrow$ source is active, if $i = 0 \Rightarrow$ source is idle.

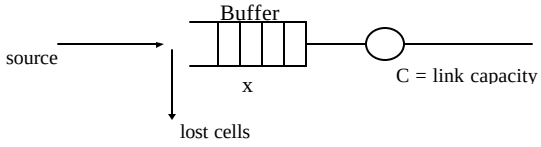


Figure 1: *ATM Multiplexer*.

Using the fluid flow model for the active periods of the source in the analysis of the buffer occupancy, the probability $P_i(t,x)$ can be found.

Supposing a buffer length $K$, the overflow probability $\varepsilon$ (the required QoS) is given by

$$\varepsilon = \beta \cdot \exp\left( -\frac{K(c - rR)}{b(1-r)(R-c)c} \right), \tag{1}$$

where

$$\beta = \frac{(c - rR) + \varepsilon\, r(R-c)}{(1-r)c}, \tag{2}$$

The effective bandwidth, $c$, for a given source can be evaluated for a given $\varepsilon$.

The Eq. 1 has no explicit solution and can only be solved numerically. Thus, the following approximation is suggested in [1].

Considering $\beta = 1$, the effective bandwidth can be evaluated and is given by [1]

$$c = \frac{a - K + \sqrt{(a-K)^2 + 4Kar}}{2a} R, \tag{3}$$

where $a = \ln(1/\varepsilon)b(1-r)R$.

For $N$ sources, the effective bandwidth is given by

$$C_E = \sum_{i=1}^{N} \hat{c}_i \tag{4}$$

where $\hat{c}_i$ is the effective bandwidth of the source $i$.

Since the effect of the aggregation of the sources is not considered in Eq. 3, the bandwidth $C_E$ can be overestimated. Thus, it is also considered an aggregated gaussian source.

For the gaussian source, the effective bandwidth is given by

$$C_G = m + \alpha' \sigma, \tag{5}$$

where,

$\alpha' = \sqrt{-2\ln(\varepsilon) - \ln(2\pi)}$,

$m = \sum_{i=1}^{N} m_i$, is the total average bit rate,

$\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$; $\sigma_i^2$ is the variance of the bit rate of the $i$th source.

The chosen bandwidth is given by

$$C_F = \min\{C_E, C_G\}. \tag{6}$$

If $C_F \le C_T$, where $C_T$ is the total link capacity, the connection is accepted, otherwise, it is rejected.

### *B. KWC Algorithm*

Kesidis, Walrand and Chang [2] proved the existence of effective bandwidth for a more general class of sources. The mathematical model is based on the assumption that the probability of the buffer occupancy obeys an exponential decay:

$$\mathrm{P}\{X > K\} \;\le\; e^{-K\delta},$$

where $\delta$ is a positive constant called asymptotic decay rate.

Furthermore, the model is based on the existence of the function

$$h(\delta) = \lim_{t \to \infty} \frac{\ln \mathrm{E}\{\exp(A(t)\delta)\}}{t} \tag{7}$$

where $A(t)$ is the number of generated cells during the time interval [0, t] and $\delta$ is a real value.

The effective bandwidth is given by

$$c(\delta) = \frac{h(\delta)}{\delta}. \tag{8}$$

The evaluation of $h(\delta)$ is in general difficult. However, for the fluid flow markovian source (on-off) the equations above can be solved and the effective bandwidth is given by

$$c = \alpha + \sqrt{\alpha^2 + \beta^2} \tag{9}$$

where [7],

$$\alpha = \frac{1}{2}\left( R - \frac{1}{\delta b} - \frac{1}{\delta T_{off}} \right), \text{ and} \tag{10}$$

$$\beta = \frac{R}{\delta T_{off}}. \tag{11}$$

$T_{off}$ is the average time of the source silent periods,

$$T_{off} = \frac{b(1-r)}{r}. \tag{12}$$

The value of $\delta$ is given by

$$\delta = \frac{\ln(1/\varepsilon)}{K}, \tag{13}$$

where $\varepsilon$ is the required cell loss probability.

### *C. SS Algorithm*

The method is based on a group of $n$ connections served by a link with capacity $C'$ bps, a buffer with length

$K$ and $p_i^*(j)$ is the arrival probability of $j$ cells from connection $i$ during the measuring interval $s$. Supposing $C = C's / L \leq K + 1$, where $L$ is the cell length in bits, the cell loss probability $\varepsilon$ is upper-bounded by

$$\varepsilon \leq \frac{\sum_{k=0}^{\infty} [k - C]^+ \, p_1^* * \cdots * p_n^*(k)}{\sum_{k=0}^{\infty} k p_1^* * \cdots * p_n^*(k)} \qquad (14)$$

where

$$[x]^+ = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

and $*$ represents convolution.

A user requests a new connection providing the peak bit rate $\chi_{n+1}$ and the mean bit rate $\gamma_{n+1}$. In order to update the $\varepsilon$ evaluation for the acceptance decision the vector $\boldsymbol{q}$ is defined representing the most bursty cell arrival distribution based on the parameters $\chi_{n+1}$ and $\gamma_{n+1}$ [3]. The vector is evaluated by

$$\boldsymbol{q}_{n+1}(j) = \begin{cases} a_{n+1} / R_{n+1}, & j = R_{n+1} \\ 1 - (a_{n+1} / R_{n+1}), & j = 0 \\ 0, & \text{otherwise} \end{cases} \qquad (15)$$

where $R_{n+1} = \lceil \chi_{n+1} s / L \rceil$ e $a_{n+1} = \gamma_{n+1} s / L$. $\lceil x \rceil$ represents the smallest integer more than or equal to $x$. $R_{n+1}$ and $a_{n+1}$ represent the maximum and mean numbers of cells arriving from connection $n+1$ during the time interval $s$. Using the traffic parameters of the requested connection $\varepsilon$ can be determined by

$$\varepsilon \leq \frac{\sum_{k=0}^{\infty} [k - C]^+ \, p_1^* * \cdots * p_n^* * \boldsymbol{q}_{n+1}(k)}{\sum_{k=0}^{\infty} k p_1^* * \cdots * p_n^*(k) \quad + \quad a_{n+1}} . \qquad (16)$$

It is possible to obtain an estimation of $p_1^* * \cdots * p_n^*(k)$ through measuring the arriving traffic. One can estimate the cell loss probability after the acceptance of the new call $\hat{\varepsilon}_1$ using the vector $\hat{\mathbf{p}} = (\hat{p}(0), \hat{p}(1), \cdots)$

$$\hat{\varepsilon}_1 \leq \frac{\sum_{k=0}^{\infty} [k - C]^+ \, \hat{p} * \boldsymbol{q}_{n+1}(k)}{\sum_{k=0}^{\infty} k \hat{p}(k) \quad + \quad a_{n+1}} \qquad (17)$$

$\hat{\varepsilon}_1$ represents the upper bound of the cell loss probability when the errors are small enough. The call is accepted only if $\hat{\varepsilon}_1$ is smaller than the required CLR.

This algorithm counts the number of cells that arrive during a measurement interval $s$ to estimate the probability distribution of cell arrivals. A set of $N$ measurement intervals is called a renewal period. The frequency distribution during the $N$ intervals inside the $t$th renewal period is estimated and represented by $\{q(k; t), k = 0, 1, \cdots\}$ and the mean value of the number of arrived cells is

$$b(t) = \sum_{k=0}^{\infty} k q(k; t) \qquad (18)$$

The vector $\hat{\mathbf{p}}$ and the scalar $\hat{a}$ must be updated every renewal period. This operation depends on the occurrence of acceptance of a new connection or releasing of an already established one. If none of them happens the estimations are updated by

$$\hat{\mathbf{p}}(t+1) = \alpha \, \mathbf{q}(t) + (1 - \alpha) \, \hat{\mathbf{p}}(t) \qquad (19)$$

$$\hat{a}(t+1) = \alpha \, b(t) + (1 - \alpha) \, \hat{a}(t) \qquad (20)$$

where $\alpha$ is a value between 0 and 1, which defines the weight of the measured traffic during the last interval.

When a new connection is established in the $t$th renewal period, it is necessary to use the traffic descriptors of the requested call and, thus, the vector $\boldsymbol{q}$ is included in the evaluation.

$$\hat{p}(k; t+1) = \hat{p}(\cdot\,; t) * \boldsymbol{q}_{n+1}(k), \quad (k = 0, 1, \cdots) \qquad (21)$$

$$\hat{a}(t+1) = \hat{a}(t) + a_{n+1} \qquad (22)$$

When a connection is released the estimations are updated as shown below:

$$\hat{p}(0; t+1) = \sum_{k=0}^{a'} \hat{p}(k; t) \qquad (23)$$

$$\hat{p}(k; t+1) = \hat{p}(k + a'; t) \qquad (k \geq 1) \qquad (24)$$

$$\hat{a}(t+1) = \hat{a}(t) - a' - \sum_{k=0}^{a'} (k - a') \, \hat{p}(k; t) \qquad (25)$$

where $a' = \hat{a}(t) - \sum_{i=1}^{n} a_i$ .

## III. Simulation and Result Analysis

### *A. Simulation*

To compare the performance of the three algorithms presented in section 2 several simulation were carried out. An ATM multiplexer handling the aggregated traffic of identical sources was used. The multiplexer was modeled as a finite buffer with capacity $K$, FIFO service discipline and one server with constant service rate. Two-state Markov fluid sources (on-off) were used. They can be characterized by their peak rate $R$, average burst length $b$ and load $r$.

It was used a link with a bandwidth of 155.52 Mbps and the required cell loss ratio, $\varepsilon$, was set at $10^{-3}$. The traffic was applied to the buffer during 900 seconds, which represents a considerable amount of information passing through the network. Before these traffic period, there is a warm-up phase in order to reach a stationary

state of the buffer occupancy.

In order to analyze the behavior of the algorithms in relation to the variations of buffer capacity, average burst length and source load the simulation was divided into three groups.

An additional group of simulations was executed to determine the upper bound on the number of connection acceptance within QoS guaranteeing. Through successive adjusts the number of accepted connections was increased or decreased until the obtained cell loss ratio was as close as possible, but lower than its required value. The results of this group of simulations represent the maximum number of connections the multiplexer can handle for the specified CLR. The behavior of the algorithms is compared to this upper bound.

The SS method uses three parameters in the process of estimating the actual traffic: the length of the measurement interval, the length of the renewal period and $\alpha$, used in the exponential forecasting as shown in Eqs. 19 and 20. The length of the renewal period is given as an amount of measurement intervals, which is defined as a number of cell slots. The length of the renewal period was equal to the length of the buffer. In all of the simulations the measurement interval length was 10,000 cell slots and the parameter $\alpha$ was set to 0.2.

The process of call arrival was simplified. During each renewal period only one call arrives at the multiplexer. This distribution makes the measured traffic more significant in the estimation of $\hat{p}$. Each requested connection is represented by Eq. 15, which means a worst case scenario for the connection. If many calls arrive at a renewal period, the acceptance decision would be strongly influenced by Eq. 15, instead of being mainly influenced by the measured traffic $\hat{p}$ (Eq. 19).

Calls are accepted until there is not enough bandwidth. After this, no more calls are generated and the simulation continues for a certain time, in order to reach a steady state. This finishes the warm-up period, which lasts a fixed number of renewal periods.

The estimation process is a time consuming operation. This is the reason why calls were distributed throughout the warm-up period and only one call arrived during any renewal period. This implies that the effect of arriving more than one connection during a renewal period was not considered in this work. This is an important point of study, because it represents the capacity the method has to react to rapid changes in the applied traffic. The effect of connection releasing was not studied here, either.

### *B. Results and Discussion*

In the first set of simulations it can be seen the effect of the variation of the buffer length on the response of the algorithms. For the performance study of the algorithms in relation to the variation of buffer length, the following parameters of source were used: peak rate = 8 Mbps, average burst length = 100 cells and load = 0.2. It was used low load to explore the statistical gain.

It can be seen in Fig. 2 that the number of connections accepted by the three algorithms is considerable different when the buffer is small ($K$ < 1,000 cells). The two effective bandwidth methods accept the same number of connections when the buffer is greater than 700 cells. The GAN algorithm presents a better link usage. Increments in the buffer size beyond 5,000 cells have little effect. Obviously, this point varies with the burst length. In the SS method, the buffer length has a smaller effect on the performance. This leads to a better performance when the buffer is small and a worse performance when the buffer is greater.

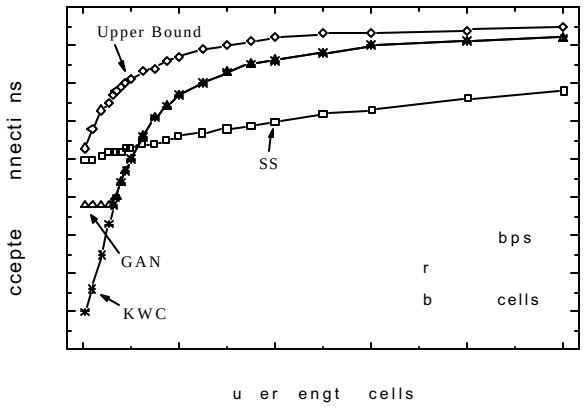


Figure 2: Number of accepted connections as a function of the buffer length.

The link utilization is considerable only when the buffer has great capacity (Fig. 3). Buffer sizes below 1,000 cells do not permit high utilization.

The Gaussian approximation used by the GAN method confirms its utility in the regions where the effective bandwidth solution overestimates the necessary bandwidth.

A capacity of 2,000 cells is sufficient to achieve high link utilization, approximately 0.8. As already seen, the SS proposal is better only for small buffers.

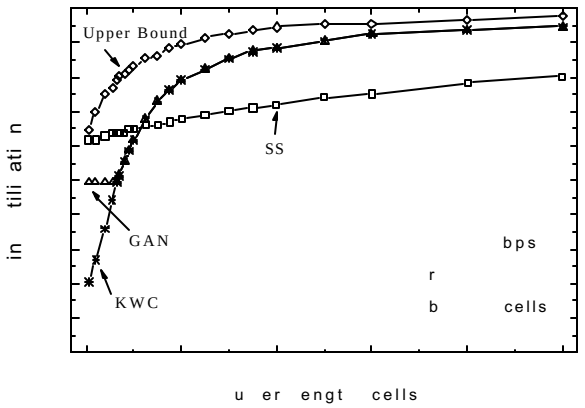


Figure 3: Link utilization as a function of the buffer length.

The cell delays obtained (Fig. 4) were very short when the buffer was also small ($K$ < 2,000 cells). Beyond this point, delay grows rapidly. Greater buffers permit a better link utilization, although they imply longer delays.

It can be noticed that link utilization near 0.6, achieved by the SS method, results in cell delay not significant. This is desirable in real time services, like voice over packet networks. Any increment in the utilization beyond 0.6 results in higher increments in cell delay, reaching undesirable delays quickly.

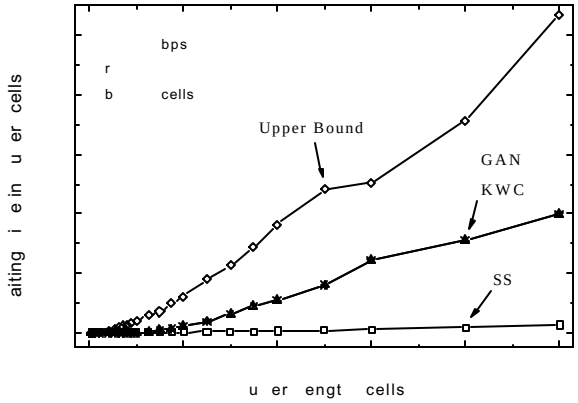


Figure 4: Average waiting time as a function of the buffer length.

The second set of simulations analyzes the behavior of the algorithms when the average burst length is changed. The sources have 8 Mbps of peak rate, buffer capacity of 4,000 cells and the generated traffic has a load of 0.3. The algorithms KWC and SS show little sensitivity to the burst length in a wide range (Fig. 5). The SS method has better performance above 1000 cells, but it is worse below this point.

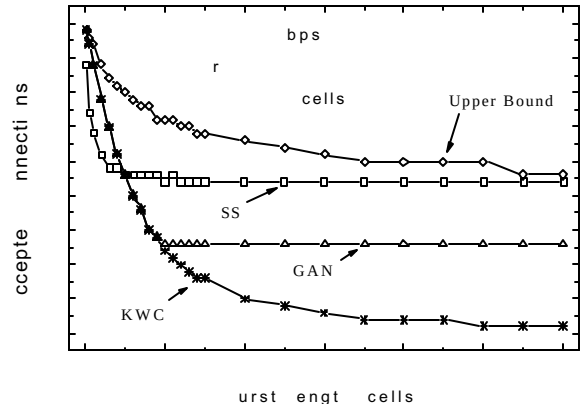


Figure 5: Number of accepted connections as a function of the burst length.

The link utilization decays rapidly with the increasing in the burst length (Fig. 6), although it can be high when bursts are shorter than 500 cells. Obviously, the buffer capacity contributes strongly to this decay. A larger buffer would result in a smoother decay.

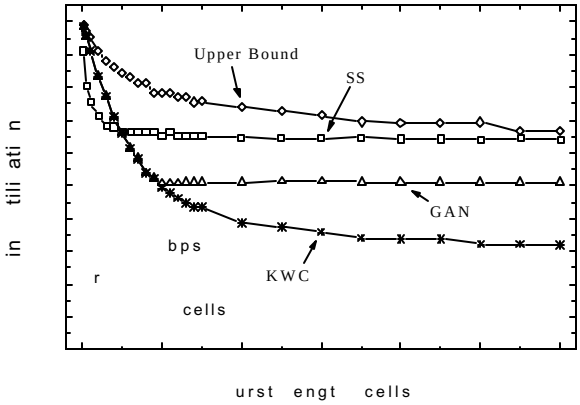


Figure 6: Link utilization as a function of the burst length.

The average waiting time is long when the link presents high utilization (Fig. 7). If the waiting time is a critical variable, it is necessary to use the link on low utilization.

By the graph, it is observed short delay, which does not represent any problem, except for small bursts, when longer delays occur.

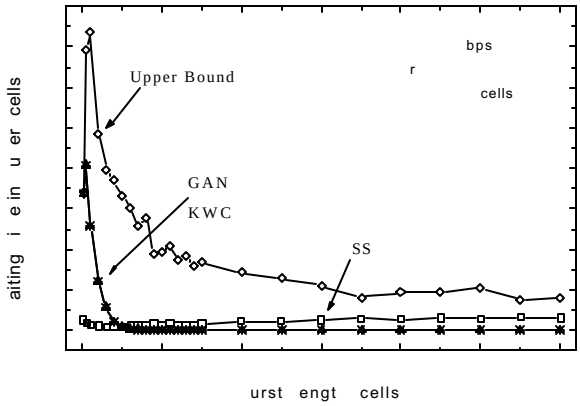


Figure 7: Average waiting time as a function of the burst length.

The third set of simulations compares the performance of the studied algorithms in relation to the average burst length. It were used sources with a peak rate of 8 Mbps, buffer capacity of 1,000 cells and an average burst length of 200 cells.

The KWC proposal is strongly conservative (Fig. 8) when the source load is low. Additional simulations showed that this difference diminish following increases in K/b ratio.

The SS proposal is more efficient for source load up to 0.4, but is slightly inferior in higher utilization.

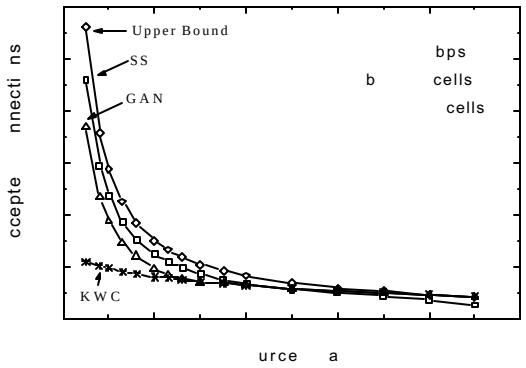


Figure 8: Number of accepted connections as a function of the source load.

In order to obtain a reasonable utilization it should be used a buffer sufficiently large to store aggregated bursts. It can be seen that low utilization happened in the region with low source load (Fig. 9). The buffer used was not large enough, which represented a constraint in the number of accepted connections. However, this limitation guaranteed very short delays (Fig. 10).

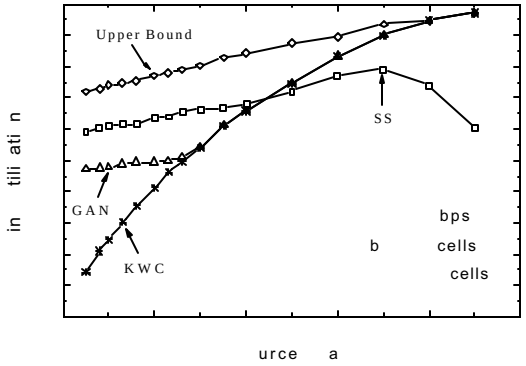


Figure 9: Link utilization as a function of the source load.

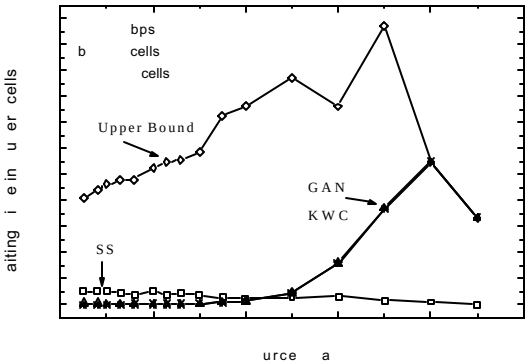


Figure 10: Average waiting time as a function of the source load.

## IV. CONCLUSIONS

Three algorithms for connection admission control (CAC) for ATM networks were studied by simulation. A simple multiplexer with only one link model was used. The study has concentrated to find the maximum number of accepted connections, which satisfy a given QoS.

The study considered the variations on buffer capacity, burst size and source load.

All the algorithms are conservative. The KWC proposal presented the worst performance. The SS method achieves good results in several situations, but showed low sensitivity to the buffer size, causing low performance in these configurations.

## REFERENCES

[1] GUÉRIN, R., AHMADI, H., and NAGHSHINEH, M.: 'Equivalent capacity and its application to bandwidth allocation in high-speed networks', *IEEE J. Select. Areas Commun*., 1991, vol. 9, no. 7, pp. 968-981

[2] KESIDIS, G., WALRAND, J., and CHANG, C.-S.: 'Effective bandwidths for multiclass Markov fluids and other ATM sources', *IEEE Trans. Networking*, 1993, vol. 1, no. 4, pp. 424-28

[3] SAITO, H. and SHIOMOTO, K.: 'Dynamic call admission control in ATM Networks', *IEEE J. Select. Areas Commun*., 1991, vol. 9, no. 7, pp. 982-989

[4] HSU, I., and WALRAND, J.: 'Dynamic bandwidth allocation for ATM switches', *J. Appl. Prob*., 1996, vol. 33, pp. 758-771

[5] LIU, K., PETR, D. W., and BRAUN, C., 'A measurement-based CAC strategy for ATM networks', *Proceedings of ICC´97*, IEEE, 1997, vol. 3, pp. 1714-18

[6] LE BOUDEC, J.-Y. and NAGARAJAN, R., 'A CAC algorithm for VBR connections over a VBR trunk', *Proceedings of 15th International Teletraffic Congress – ITC*, 1997, vol. 1, pp. 59-70, Washington, DC, USA

[7] OLIVEIRA, J. C., BONATTI, I. S., PERES, P. L. D. and BUDRI, A. K., 'Cell level performance approach for link dimensioning of ATM networks', *Proceedings of ITS 98*, 1998, vol.1, pp. 189-194, São Paulo, SP, Brazil.