

Efficient Estimation of Cell Loss Probability in ATM multiplexers with a Fuzzy Logic System*

Shirley T. C. Lam, Brahim Bensaou and Danny H. K. Tsang

Hong Kong University of Science and Technology

Department of Electrical and Electronic Engineering

Clear Water Bay, Kowloon, Hong Kong.

E-mail: {lamtc, eebben, eetsang}@ee.ust.hk

fax: (852)-2358-1485

Abstract

An important parameter in ATM-based network design and management is the cell loss probability in ATM multiplexers. However, it depends on many unknown and unpredictable traffic parameters such as burst length distribution, mean rate, etc. In this paper, we propose a simple and robust fuzzy-based algorithm to predict the cell loss probability in large-sized systems based only on both a small amount of information from small-sized systems, and the asymptotic behavior for very large systems. Numerical results show that the value predicted by this algorithm is quite accurate. The application of the proposed algorithm to call admission control is also presented.

Keywords

ATM, cell loss probability, fuzzy logic, call admission control.

1 INTRODUCTION

An ATM-based Broadband Integrated Services Digital Network (B-ISDN) is a single high speed network designed to support all known and unknown services with different quality of service (QoS) requirements. It transfers the information through fixed size packets called cells. The network takes advantage of the statistical behavior of the sources with different traffic characteristics to efficiently share transmission resources through statistical multiplexing. An ATM multiplexer consists of a buffer of size K (cells) and a single output link with a transmission capacity of C cells per second. When the input rate is greater than the transmission capacity and the buffer is full, cell loss occurs. Cell loss has a harmful effect on the QoS. Therefore, an accurate estimation of the cell loss probability

*Supported by HongKong Telecom Institute of Information Technology grant HKTIIT93/94. EG01

(CLR[†]) not only gives a good estimation of this QoS but also provides an important parameter needed for network design and management (e.g., buffer dimensioning, congestion control, call admission control and routing).

Many approaches for evaluating the CLR at the so-called burst-scale congestion for multiplexers loaded with a superposition of on/off sources have been proposed in the literature. The first approach approximates the actual arrival process by fluid flow (FF) (see Anick et al. (1982) and Bensaou et al. (1994)). In this approximation, the fluctuation of cell arrival rates can accurately be represented by assuming that the information arrives in a continuous flow rather than in discrete cells. The CLR is accurately approximated by the overflow probability which is obtained by solving an adequate eigensystem (Anick et al. (1982)). Another approach, very similar to the fluid flow method, approximates the actual arrival process by a Markov modulated deterministic process (MMDP) in which cells arrive according to a deterministic renewal process whose rate is controlled by a Markov process as discussed by Yang and Tsang (1995). In this approach, the exact cell loss probability is obtained by solving a set of linear equations through the Gauss-Seidel algorithm.

These approaches are efficient in predicting the CLR in an ATM multiplexer. However, when the system size becomes large, computation complexity increases in the FF approximation and memory problems arise in the MMDP approximation. It is also shown in Bensaou et al. (1994) that the CLR depends on many unknown and unpredictable traffic parameters (e.g., burst length distribution,...). To avoid these problems, the goal is to derive a model-independent algorithm to predict the CLR in large-sized systems by relying only on information from some small-sized systems.

In this spirit, the Global Rational Approximation (GRA) algorithm is proposed by Yang et al. (1995) to approximate the CLR. In this method, the CLR as a function of buffer size (or of number of users) is approximated by a rational function, $R(x) = P_m(x)/Q_n(x)$ where P_m and Q_n are polynomial functions of degree m and n respectively. The coefficients of P_m and Q_n are determined by solving the system of linear equations: $R(x_i) = \text{Pr}(x_i)$, $i = 1, \dots, m+n$; where the pairs $(\text{Pr}(x_i); x_i)$ represents the small-sized system information (e.g., (CLR; buffer size), (CLR; number of users)).

Even though the GRA approximation is efficient and accurate in many cases, it still has some major problems:

- the accuracy of the approximation is closely related to the degrees m and n , which in turn determine the number of pairs $(R(x_i); x_i)$ required (i.e., $m+n$ pairs). When the degrees are not large enough, the accuracy of GRA can be poor;
- computation time can sometimes be quite long since the fitting process is based on an iterative algorithm to determine the suitable values of m and n for the required accuracy. In addition, depending on the values of the pairs $(R(x_i); x_i)$, the system of equations to solve sometimes can encounter singularity problems, which make the fitting process difficult. This problem occurs mainly when the known values of CLR do not fit on a smooth curve. This fact restricts the practical use of GRA when the known values of CLR are obtained from measurements that normally contain a small amount of error.

[†]in the sequel we use the generic term cell loss probability or CLR to designate either the cell loss probability, the overflow probability or the cell loss ratio

Fuzzy logic technique has been proposed to efficiently solve several ATM problems (e.g. Chang and Cheng (1994) and Cheung et al. (1994)). Thus, in view of the shortcomings encountered in the above mentioned approaches, we propose to use an adaptive fuzzy system to efficiently predict the CLR in large-sized systems based on only a small amount of information from small-sized systems. This information can be obtained by either real-time measurement when the traffic characteristics are unknown or from any analytic model when the traffic characteristics are known. This predicted CLR is then refined, by taking advantage of the knowledge of the asymptotic behavior of the performance curve (CLR). The advantages of using our proposed fuzzy logic based algorithm include robustness, simplicity in implementation, computational efficiency, and good accuracy.

In Section 2, we present a brief introduction to an adaptive fuzzy system and then describe our proposed fuzzy approximation (FA) algorithm. In Section 3, we give some numerical results to validate our algorithm. The FA algorithm is used to predict the CLR as a function of buffer size. The results are compared to those obtained from other alternative methods. In Section 4, we show through a simple example how this algorithm can be used to develop a call admission control solution in ATM networks. The algorithm is used to predict the CLR as a function of the number of connected users for both homogeneous and heterogeneous traffic. We finally draw our conclusion in Section 5.

2 FUZZY APPROXIMATION

In this section, we briefly introduce an adaptive fuzzy system and show how it can be used to represent an unknown mathematical model. We will then show how this system is applied to predict the CLR in a large-sized system by taking information from both the small-sized systems and the asymptotic behavior of the performance curve for the large-sized system.

2.1 Fuzzy Rule-based System

Suppose we have a system which is almost impossible to be represented accurately by any analytical model. And suppose we know N sets of input-output pairs $(\mathbf{X}_0^j; y_0^j)$, $\mathbf{X}_0^j = (x_{01}^j, \dots, x_{0n}^j)^T \in \mathbf{R}^n$, $j = 1, 2, \dots, N$, where N is a small number. A general method is provided by Wang (1994) to design a fuzzy system which can match all the N input-output pairs to any given accuracy (i.e., for any given $\epsilon > 0$, we require that $|f(\mathbf{X}_0^j) - y_0^j| < \epsilon$ for all $j = 1, 2, \dots, N$). This optimal fuzzy system is constructed as

$$f(\mathbf{X}) = \frac{\sum_{j=1}^N y_0^j \prod_{i=1}^n \mu_{\mathbf{A}_i^j}(x_i)}{\sum_{j=1}^N \prod_{i=1}^n \mu_{\mathbf{A}_i^j}(x_i)}, \quad (1)$$

where \mathbf{A}_i^j are fuzzy sets defined by their membership functions $\mu_{\mathbf{A}_i^j}$, $i = 1, \dots, n$ and $j = 1, \dots, N$.

The above fuzzy system is constructed from N IF-THEN rules which have the following form:

$$\text{IF } x_1 \text{ is } \mathbf{A}_1^j \text{ and } \dots \text{ and } x_n \text{ is } \mathbf{A}_n^j \text{ THEN } y \text{ is } \mathbf{B}^j, \quad (2)$$

where \mathbf{A}_i^j and \mathbf{B}^j are fuzzy sets in \mathbf{R} , and $\mathbf{X} = (x_1, x_2, \dots, x_n)^T \in \mathbf{R}^n$ and $y \in \mathbf{R}$ are respectively the input and output variables of the fuzzy system. The membership function of \mathbf{A}_i^j is defined as $\mu_{\mathbf{A}_i^j}(x_i) = \exp\{-(x_i - x_{0i}^j)^2 / (\sigma_i^j)^2\}$ where σ_i^j is used to control the accuracy of the matching and the center of \mathbf{B}^j equals to y_0^j . By appropriately choosing the parameters σ_i^j , the fuzzy system in (1) can match all the N input-output pairs to any given accuracy ϵ .

As mentioned by Wang (1994), the fuzzy system defined in (1) is exactly the same as the probabilistic general regression formula.

2.2 Application to CLR estimation

The fuzzy approximation (FA) we propose to estimate the cell loss probability in ATM multiplexers is based on the above fuzzy system. We view the CLR as an unknown function of a variable which can be the multiplexer's buffer size, the number of connected users, or the service capacity.

The algorithm we propose to predict the CLR does not assume anything about the traffic parameters but requires only the knowledge of

- the CLR of the multiplexer when the system size is small (e.g., small buffer size). In this case, the CLR is relatively large and can be calculated or measured quickly and easily;
- the asymptotic behavior of the CLR when the system size is very large (e.g., infinitely large buffer size or number of users).

Let I be the number of known values from the small-sized system. Let $P(m)$ be the cell loss probability when the system is in state S_m , where S_m can be the buffer size B_m or the number of users N_m . The set of states (S_m) should define a linearly increasing function of the form: $S_m = m\delta + S_0$, where $\delta > 0$ is the step size and S_0 is the initial state. Denote by k the index of the current state S_k .

Based on the above informations, the FA algorithm is constructed as follows:

Step 1: Input-Output Pairs Definition

The $(i - 2)$ sets of input-output pairs are chosen as follows:

$$\begin{aligned} & \{(P(k-2), P(k-1)); P(k) - P(k-1)\}, \\ & \{(P(k-3), P(k-2)); P(k-1) - P(k-2)\}, \\ & \vdots \\ & \{(P(k-(I-1)), P(k-(I-2))); P(k-(I-3)) - P(k-(I-2))\}, \end{aligned} \quad (3)$$

where $(P(k-(j+1)), P(k-j))$ is the input vector and $P(k-(j-1)) - P(k-j)$ is the output for $j = 1, \dots, I-2$.

By taking advantage of the monotonicity of the parameterized curve, representing the

CLR as function of the buffer size or the number of users, the fuzzy system can be constructed from only a few input-output pairs. The inputs are represented by the CLR values from the previous and the current states and the output is the difference between the next and the current CLR values. The input-output pairs are updated in real time.

Step 2: Rule Base Generation

For each input-output pair, one fuzzy rule base is generated according to (2). The rules in the fuzzy rule base become

$$\begin{aligned}
 &\text{IF } P(k-1) \text{ is } \mathbf{A}_1^1 \text{ and } P(k) \text{ is } \mathbf{A}_2^1 \text{ THEN } (P(k+1) - P(k)) \text{ is } \mathbf{B}^1, \\
 &\text{IF } P(k-1) \text{ is } \mathbf{A}_1^2 \text{ and } P(k) \text{ is } \mathbf{A}_2^2 \text{ THEN } (P(k+1) - P(k)) \text{ is } \mathbf{B}^2, \\
 &\vdots \\
 &\text{IF } P(k-1) \text{ is } \mathbf{A}_1^{I-2} \text{ and } P(k) \text{ is } \mathbf{A}_2^{I-2} \text{ THEN } (P(k+1) - P(k)) \text{ is } \mathbf{B}^{I-2},
 \end{aligned} \tag{4}$$

where the membership functions of the input fuzzy sets are given by

$$\begin{cases} \mu_{\mathbf{A}_1^j}(P(k-1)) &= \exp \left\{ -\frac{(P(k-1) - P(k-j-1))^2}{(\sigma_1^j)^2} \right\}, \\ \mu_{\mathbf{A}_2^j}(P(k)) &= \exp \left\{ -\frac{(P(k) - P(k-j))^2}{(\sigma_2^j)^2} \right\}, \end{cases} \quad j = 1, \dots, I-2, \tag{5}$$

with

$$\begin{cases} \sigma_1^j &= \frac{\max_j (P(k-j-1)) - \min_j (P(k-j-1))}{I-2}, \\ \sigma_2^j &= \frac{\max_j (P(k-j)) - \min_j (P(k-j))}{I-2}, \end{cases} \quad j = 1, \dots, I-2, \tag{6}$$

and the center of the fuzzy set \mathbf{B}^j is $P(k-j+1) - P(k-j)$, $j = 1, \dots, I-2$.

Note that σ_1^j and σ_2^j , $j = 1, \dots, I-2$, are free parameters that determine the accuracy of the approximation. The above choice of σ_1^j and σ_2^j , $j = 1, \dots, I-2$, makes the membership functions uniformly cover the range of the inputs vectors.

Step 3: Fuzzy system construction

The fuzzy system given in (1), with $n = 2$ and $N = I - 2$, is chosen to estimate the increment (or decrement) of the CLR. The output of the fuzzy system will be added to (or subtracted from) $P(k)$ to estimate $P(k+1)$. This predicted value of $P(k+1)$ is fed back into the fuzzy system as an input to estimate $P(k+2)$ using $P(m)$, $m = k - N, \dots, k+1$ and the process continues until we get the CLR for the desired size of the system.

Step 4: Use of the asymptotic information

Generally, the FA algorithm provides an accurate prediction when the number of states to predict is not too large. Nevertheless, predicting CLR values of the order of 10^{-10} requires a large number of states, this makes the accuracy of the prediction poor because of the accumulation of small errors. In order to improve the estimation, we have to take into account the asymptotic behavior of the CLR as the system size S_m becomes infinitely large.

When S_m is the multiplexer's buffer size (B_m), we know that the CLR behaves like (Anick et al. (1982))

$$P(m) \sim L_0 \exp\{-\eta B_m\}, \quad \text{as } B_m \rightarrow \infty, \quad (7)$$

where η is given in Anick et al. (1982). In other words, we have $\log P(m+1) - \log P(m) \sim -\eta(B_{m+1} - B_m) = -\eta\delta$, for $B_m \rightarrow \infty$, where δ is the step increment in buffer size.

When S_m is the number of users (N_m), we know that

$$P(m+1) - P(m) \sim 0, \quad \text{whenever the system load } \geq 1. \quad (8)$$

By taking the advantage of this additional knowledge, the predicted CLR value becomes

$$P(k+1) = (1 - \lambda(k))P_1(k+1) + \lambda(k)P_2(k+1), \quad (9)$$

where $\lambda(m)$ is a nonlinear increasing function of S_m taking its values in $[0, 1]$, $P_1(k+1)$ is the predicted value in Step 3 above and $P_2(k+1)$ is the predicted value based on the above asymptotic behavior. Finally, we feed this predicted value to the fuzzy system in Step 1 to predict the next value.

The function $\lambda(m)$ we use is given by

$$\lambda(m) = \begin{cases} \exp\left(-\frac{(S_m - S_\infty)^2}{\sigma^2}\right) & \text{if } S_m \leq S_\infty \\ 1 & \text{elsewhere,} \end{cases} \quad (10)$$

where S_∞ is any state at which the asymptotic behavior begins to hold and σ is a parameter controlling the accuracy of the fuzzy approximation. The details of determining S_∞ and σ will be discussed later in Section 3.3.

3 NUMERICAL RESULTS

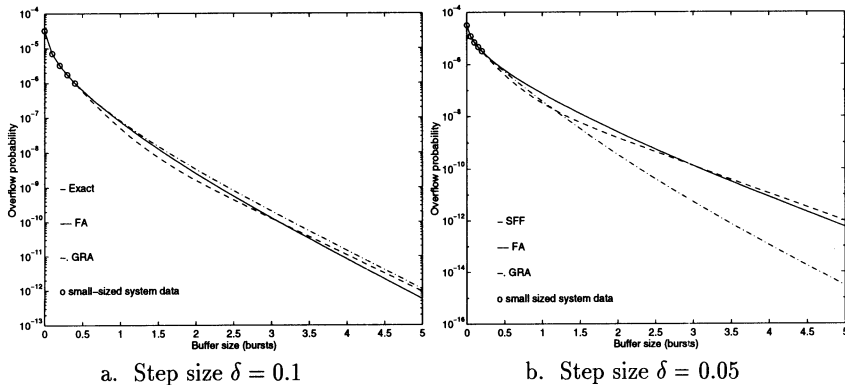
The implementation of the algorithm described above proves straightforward, and the CPU time needed to predict the CLR is quite small. To illustrate the accuracy and the robustness of the FA algorithm, we present in the following a sample of figures representing the CLR as a function of the buffer size. On this figures, our results are compared to the other alternative methods presented in the literature. The model we use to derive the "exact" results is based on the FF approximation with different burst length distributions as discussed in Bensaou et al. (1994). The characteristics of the input traffic are depicted in Table 1.

3.1 Validation of the FA approximation

To show the accuracy and the effectiveness of the FA algorithm, we use the minimum necessary set of input/output pairs. In Figure 1, this number is equal to five ($I = 5$). This means that the fuzzy rule base contains three IF-THEN rules. The figure shows the overflow probability as a function of the buffer size for voice calls (peak rate = 64

Table 1 Traffic Characteristics used.

Traffic class	Peak Rate (Mb/s)	Mean Rate(Mb/s)	Burst length (cells)
Data	10	1	339
Image	2	0.087	2604
Voice	0.064	0.022	58

**Figure 1** Voice calls ($C = 1.5$ Mb/s, $N = 34$).

kb/s, mean rate = 22 kb/s, mean burst duration = 350 ms). The system load is 0.5 and the number of multiplexed sources is equal to 34. In Figure 1.a and Figure 1.b, the five input/output pairs are obtained from the FF approximation with a step size (δ) of 0.1 and 0.05 (mean burst length) respectively, beginning at 0. As shown in the figures, our fuzzy algorithm gives a very good approximation compared to the exact results and to those provided by GRA. The CPU time (10ms) needed by the FA algorithm is, however, 10 times smaller than that (100ms) needed by the GRA algorithm and about 20 times less than that (200ms) needed by the exact algorithm. We mention however that the CPU time needed by the FF approach is not representative of the CPU time needed by analytic models. The above value is obtained from the only known closed-form result. For the other algorithmic solutions, this can be much larger.

3.2 Independence from a model

The fuzzy approach used to predict the CLR is to some extent independent of any analytical modelling of the actual input process. To illustrate this independence, in this section we show two kinds of figures.

The first kind (Figure 2.a) shows the overflow probability against the buffer size for generally distributed burst lengths, as discussed by Bensaou et al. (1994). To simplify the exact model, the number of sources is assumed to be infinitely large. The burst arrival

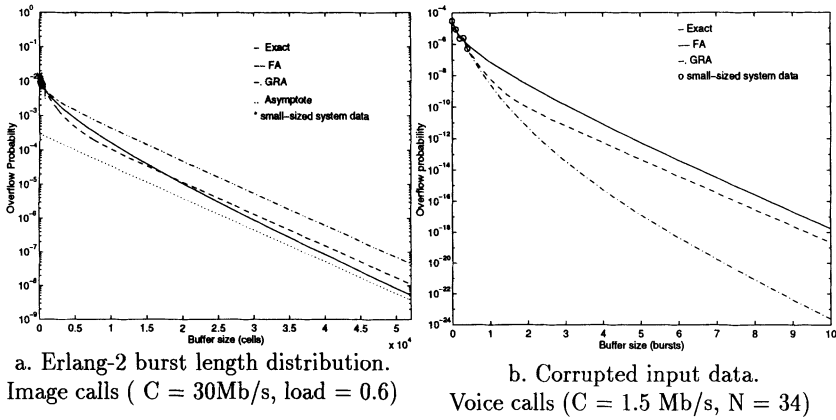


Figure 2 Independence of the FA approximation from the model.

process thus becomes Poissonian and the individual mean rate is infinitely small. This assumption does not restrict the generality of our algorithm. In Figure 2.a the burst lengths are 2-phase Erlang distributed with a mean value of 2604 cells and the peak rate is equal to 2Mbits/s . As shown by the curves, the FA algorithm gives a very reasonable approximation. The error is always within one order of magnitude. More numerical examples with other burst length distributions have shown the same accuracy. This shows that our approach is independent of any assumption on the traffic parameters such as the burst length distributions, etc.

The second kind (Figure 2.b) shows the prediction of the overflow probability from a measured set of input/output pairs with some random measurement errors. To simulate this measured set, we take the exact (FF) curve for a superposition of voice sources from Figure 1.a, and introduce a small perturbation to the first 5 points by adding (or subtracting) a value ranging from 0% to 50% respectively to (or from) these points. The smaller is the overflow probability, the larger is the introduced error. In other words, the first point has no error while the last point can have relative error as large as 50%. This simulates the situation of on-line measurement of real traffic when the measurement time is not very large. Figure 2.b shows the accuracy of the FA algorithm in predicting the overflow probability. It is clear that the refinement based on the knowledge of the asymptotic slope is very valuable in this kind of situations.

These two figures have shown the robustness and the independence of our algorithm from any specific traffic model. Moreover, the algorithm can be easily extended to the case of a superposition of heterogeneous classes of traffic without increasing the complexity. This makes it very appealing for real-time applications.

3.3 Determination of S_∞ and σ

We would like to draw the attention of the reader to the fact that the performance of the FA approximation is largely determined by the choice of the values of the parameters S_∞ and σ . In this section, the state S_m has been chosen to be the buffer size B_m . The value

for $S_\infty (= B_\infty)$ must be chosen such that for a buffer of size B_∞ the asymptotic behavior holds. A heuristic approach is to set B_∞ equal to ten times the mean burst length when the traffic intensity is light (< 0.85), while a value of a few times the mean burst lengths is sufficient in heavy traffic conditions. In the figures above, the value of B_∞ was equal to 10 mean burst lengths. In addition, extensive calculations have shown that a value of σ equal to 60% of the value of B_∞ is quite reasonable. Furthermore, we can combine our fuzzy system with other adaptive learning algorithm such as neural networks to determine the optimal values of B_∞ and σ . Therefore, one can first train the fuzzy system with off-line simulation and then use it with on-line traffic measurement.

4 APPLICATION TO CALL ADMISSION CONTROL

One of the most important procedures to control congestion in the network is the admission control process, which restricts the number of calls within the network to provide and ensure QoS guarantees to all users in progress. To allow the network to decide quickly whether a new call can be accepted or not, a fast algorithm is required to estimate either the CLR or the required bandwidth. Most of the proposed call admission control (CAC) algorithms (e.g., Guérin et al. (1991)) are based on the effective bandwidth (the asymptotic upper bound or the stationary Gaussian approximation) without taking into account of statistical multiplexing among sources. However, both algorithms overestimate the required aggregate bandwidth, resulting in an under-utilization of expensive network resources such as bandwidth and memory.

In this section, we present a simple admission control algorithm based on our FA algorithm. This CAC algorithm should just be taken as a simple example to illustrate the application of the FA algorithm to real ATM problems. The algorithm we present below is effective only when the traffic model can be known. It considers the CLR, as a function of the number of users. A more practically interesting measurement-based CAC algorithm using our FA algorithm to predict the required aggregate bandwidth is presented in the work of Chu et al. (1996).

4.1 Homogeneous traffic sources

Without loss of generality, we consider an ATM multiplexer with a high speed channel serving homogeneous sources and a buffer of finite size to bound the delay requirements. To provide QoS guarantees such as the cell loss probability (e.g., $\text{CLR} < \epsilon$), we need to be able to estimate the CLR with different number of users. Let $P(N)$ denote the CLR when the number of users is N . When N is small, we can use any proposed analytic model (such as FF, MMDP, etc.) to calculate the values of the CLR. Nevertheless, the computational complexity increases dramatically for large number of sources. As discussed in Section 2, we can easily apply our FA algorithm to predict the CLR as a function of the number of users. Therefore, we can implement our CAC according to the following steps.

1. We first obtain four initial values of the CLR $\{P(N_0), P(N_0+1), P(N_0+2)$ and $P(N_0+3)\}$ with some analytic model, where N_0 is chosen large enough to ensure that the CLR is not equal to zero. For instance, the number N_0 can be chosen such that $N_0 - 1$ is the maximum number of admissible users with peak bandwidth allocation.

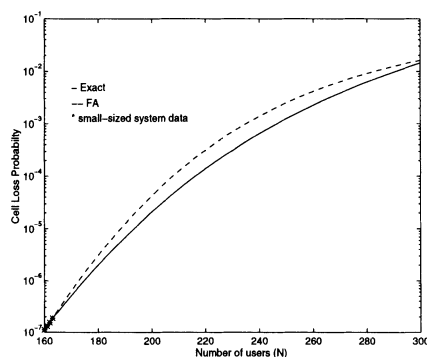


Figure 3 Cell loss probability vs. number of users (Data source, $C = 350$ Mb/s).

Table 2 Mixed traffic sources.

Traffic class	Peak Rate (Mb/s)	Mean Rate(Mb/s)	Burst duration (s)
Class 1	0.1	0.01	0.35
Class 2	0.2	0.02	0.175

- Using $\{P(N_0), P(N_0+1), P(N_0+2)$ and $P(N_0+3)\}$, the FA algorithm is used to predict $P(N_0+4)$ and then $P(N_0+k)$ for $k > 4$ successively. The asymptotic behavior is given in (8).
- We obtain the curve of CLR as a function of the number of users from Step 2. When a new user requests a connection, the call admission controller will decide to accept the new call if the resulting CLR is smaller than ϵ or to reject it if the value is greater.

In order to test the performance of the FA algorithm, Figure 3 shows the FA prediction of the CLR against the number of users. The exact model we used is based on the MMDP approximation developed by Yang and Tsang (1995). The sources are data sources with traffic characteristics given in Table 1. The service capacity is equal to 350Mb/s and the buffer can accommodate upto 50 cells. From the figure, the accuracy of this algorithm is much better than the results obtained from the effective bandwidth approach using the asymptotic FF upper bound (see Guérin (1991)). We omit this latter curve from the figure, the CLR being always close to 1. The CPU time needed is independent of the number of sources and is linearly dependent on the number of initial values.

4.2 Two heterogeneous classes of traffic sources

The simplicity of this algorithm and its speed make it very easy to extend to the heterogeneous case, for which the functions to estimate become multidimensional. To illustrate this, Figure 4 shows the relative error for the mix of two generic classes of traffic, each one contributing 50% of the total load. The channel capacity is 1.5Mb/s and the buffer can accommodate up to 2358 cells. The traffic characteristics are given in Table 2.

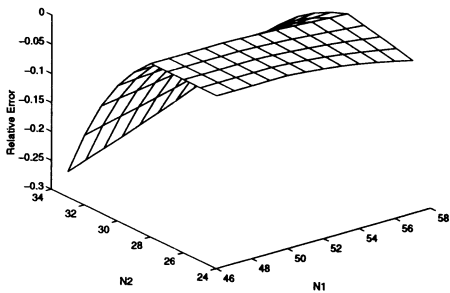


Figure 4 Prediction error of the aggregate overflow probability for 2 classes.

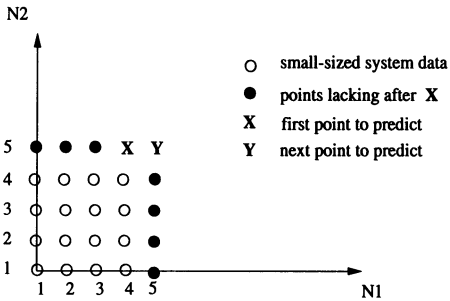


Figure 5 Prediction path for 2 classes of traffic sources.

In this case, the FA algorithm relies on the knowledge of 16 initial points (the 16 empty circles in Figure 5). From these values and knowing that the CLR converges to 1 when the number of sources is very large, we can predict the CLR surface. Since the predicted surface and the exact one are too close, we show the relative prediction error instead. The figure shows that the relative error is reasonably small (less than 30%). Extensive numerical tests have shown that the accuracy of prediction is independent of the followed path: fix the number of class 1 sources and predict towards the increasing number of class 2 sources, or vice versa. The complexity in this heterogeneous case is unfortunately dependent on the number of sources in $M - 1$ classes, where M is the total number of traffic classes. As shown in Figure 5, assume the first user to arrive is from class 2. By using the four points in column 4, the first point to predict in Figure 5 is then **X**. Assume the next user to request a connection is from class 1, from Figure 5 we see that in either direction (row or column), there is a lack of points to predict the point **Y**. We therefore need to predict at most 4 points preceding **Y** in either direction (3 points in row 5 or 4 points in column 5). Nevertheless, since the algorithm is intended for on-line use for which only one point is predicted when a new connection arrives, the prediction of the future lacking points can be done off-line after a call is accepted. This dependence on the number of sources then becomes unimportant.

The values of N_∞ and σ are very important. To obtain tight predictions, we have noticed from numerical tests that the suitable value of N_∞ is equal to 10 times the ratio of the channel capacity to the mean rate, and σ is approximately 48% of the value of N_∞ .

5 CONCLUSIONS

A new approximation algorithm is proposed to estimate the CLR for large sized systems by using a small amount of information from small-sized systems. In this algorithm, we use an adaptive fuzzy system to predict the CLR of the actual system. This prediction combined with the knowledge of the asymptotic behavior allows us to obtain a good approximation of the CLR.

To illustrate the efficiency and the practical utility of our algorithm, we have proposed one possible implementation of a CAC mechanism using the predicted CLR from the proposed FA algorithm. The numerical results have shown that the FA-based CAC algorithm outperforms the traditional approaches such as effective bandwidth without requiring excessively more CPU time than these approaches.

Currently, we are extending our work to other types of traffic such as self-similar traffic. Since our approach is independent of any traffic model, the FA algorithm is expected to predict accurately the CLR for this kind of traffic sources as well. The extension of the FA algorithm towards CAC in more complex networks composed of multiple multiplexing stages is also under examination.

REFERENCES

- Anick, D., Mitra, D., and Sondhi, M. (1982). Stochastic theory of a data handling system with multiple sources. *Bell Systems Technical Journal*, 61(8):1871–1894.
- Bensaou, B., Guibert, J., Roberts, J. W., and Simonian, A. (1994). Performance of an ATM multiplexer queue in the fluid approximation using the Beneš approach. *Annals of Operations Research*, 48.
- Chang, C. J. and Cheng, R. G. (1994). Traffic control in an ATM network using fuzzy set theory. In *IEEE Infocom'94*, pages 1200–1207.
- Cheung, K. F., Tsang, D. H. K., Cheng, C. C., and Liu, C. W. (1994). Fuzzy logic based ATM policing. In *IEEE ICCS'94*, pages 535–539.
- Chu, H.-W., Bensaou, B., Lam, S. T. C., and Tsang, D. H. K. (1996). Call admission control in ATM using fuzzy logic. Submitted.
- Guérin, R., Ahmadi, H., and Naghshineh, M. (1991). Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE JSAC*, 9(7):968–981.
- Wang, L. X. (1994). *Adaptive fuzzy system and control: design and stability analysis*. Prentice Hall, Inc.
- Yang, H., Towsley, D., and Gong, W. (1995). Efficient calculation of cell loss in ATM multiplexers. In *IEEE Globecom'95*, pages 1226–1230.
- Yang, T. and Tsang, D. H. K. (1995). A novel approach to estimating the cell loss probability in an ATM multiplexer loaded with homogeneous sources. *IEEE Transactions on Communications*, 43(1):117–126.