ELSEVIER

Computer Communications 20 (1997) 732–740

computer communications

Neural-network-based call admission control in ATM networks with heterogeneous arrivals

Jen M. Hah, Po L. Tien, Maria C. Yuang

Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, ROC

Received 12 September 1996; revised 2 March 1997; accepted 14 May 1997

Abstract

Call Admission Control (CAC) has been accepted as a potential solution for supporting diverse, heterogeneous traffic sources demanding different Quality of Services (QOSs) in ATM networks. Also, CAC is required to consume a minimum of time and space to make call acceptance decisions. In this paper, we present an efficient neutral-network-based CAC (NNCAC) mechanism for ATM networks with heterogeneous arrivals. All heterogeneous traffic calls are initially categorized into various classes. Based on the number of calls in each class, NNCAC efficiently and accurately estimates the cell delay and cell loss ratio of each class in real time by means of a pre-trained neutral network. According to our decent study which exhibits the superiority of the employment of analysis-based training data over simulation-based data, we particularly construct the training data from a heterogeneous-arrival dual-class queueing model $M^{[N_1]} + I^{[N_2]}/D/1/K$, where M and I represent the Bernoulli and interrupted Bernoulli processes, and N_1 and N_2 represent the corresponding numbers of calls, respectively. Analytic results of the queueing model are confirmed by simulation results. Finally, we demonstrate the profound agreement of our neural-network-based estimated results with analytic results, justifying the viability of our NNCAC mechanism. © 1997 Elsevier Science B.V.

Keywords: Call Admission Control (CAC); Quality of Service (QOS); Neutral network; Cell delay; Cell loss ratio; Heterogeneous-arrival queueing model

1. Introduction

Asynchronous Transfer Mode (ATM) networks [1,2] are expected to fully utilize network resources while retaining satisfactory Quality of Service (QOS) for each user in broadband-ISDNs (B-ISDNs) [3]. To satisfy this requirement, Call Admission Control (CAC) [4] is one of the potential solutions. Essentially, CAC is required to consume a minimum of time and space to make call acceptance decisions based on various QOS requirements. Numerous CAC mechanisms, which have been proposed, fall into one of three main categories: delay-based [5], loss-based [6–12], or delay-and-loss-based [13,14].

Generally, the mechanisms of the first two categories take only the delay or loss QOS into consideration. On the other hand, the mechanisms of the last category offer preferable CAC by considering both delay and loss QOSs, at the expense, however, of an increase in the time and space complexity. To cope with the problem, neural network estimation methods [15–18] have emerged and have been shown to be promising for the efficient operation of CAC in ATM networks. Furthermore, most of the existing neural-network-based CAC mechanisms, shown to exhibit various performance merits, perform the off-lined training of neural networks via simulation data. Despite the fact that the amount of training time is disregarded, owing to the off-line nature of the training, the determination of whether the simulation has converged to steady state, as will be shown, is always non-trivial especially under low-loss conditions of networks.

The major goal of the paper is to propose a highly efficient neural-network-based CAC (NNCAC) mechanism for ATM networks with heterogeneous arrivals. All heterogeneous traffic calls are initially categorized into various classes. Based on the number of calls in each class, NNCAC efficiently and accurately estimates the cell delay (CD) and cell loss ratio (CLR) of each class in real time by means of a pre-trained neural network. The second goal of the paper is to examine the superiority of the employment of analysis-based training data over simulation-based data. As a result, we particularly construct the training data from a heterogeneous-arrival dual-class queueing model, $M^{[N_1]} + I^{[N_2]}/D/1/K$, where M and I represent the Bernoulli process and the interrupted Bernoulli process (IBP), and N_1 and N_2 represent the corresponding numbers of calls, respectively. Analysis results of the queueing model are

PII S0140-3664(97)00101-1

confirmed by simulation results. Finally, we demonstrate the profound agreement of our neural-network-based estimated results with analytic results, justifying the viability of our NNCAC mechanism.

The rest of the paper is organized as follows. Section 2 first presents the analysis of the CD and CLR for the queueing system, $M^{[N_1]}+I^{[N_2]}/D/1/K$. Simulation results are provided to confirm the accuracy of the analysis. Section 3 then proposes the NNCAC mechanism in which the training data of the neural network are collected from the results obtained from Section 2. Comparisons between the NNCAC results and analytic results are also drawn. Finally, Section 4 concludes the paper.

2. Queueing model and analysis

All traffic source streams (calls) are categorized into various classes based on the mean cell arrival rate and mean burst length. That is, streams of the same class have the same mean cell arrival rate and mean burst length. In addition, any non-bursty source stream (such as files or any stream output from a traffic shaper [19] is modelled as a Bernoulli process (called the M-stream), whereas any bursty source stream (such as voice, video) is modelled as an IBP (called the I-stream) [20,21]. For simplicity, we consider the analysis of the system with one M-stream class (referred to as class-a) and one I-stream class (referred to as class-b). It is worth noting that the following dual-class analysis can be easily expanded and applied to any number of classes in the system [22]. This fact renders the employment of analysis-based training data feasible under diverse traffic loads.

For class-a, we further assume that there are N_a M-streams. These M-stream cells are referred to as M-cells. The observed M-cell is denoted as M^{O}-cell. Let Ω be the number of M-cells arriving in a slot time (where slots are fixed in length), and R the mean cell arrival rate (cells/slot time), then the probability mass function $m(j)$ of Ω becomes

$$m(j)=\mathrm{Prob}[\Omega=j]=\binom{N_a}{j}\cdot R^j\cdot(1-R)^{N_a-j},\quad 0\le j\le N_a.$$

For class-b, we assume that there are N_b I-streams. These I-stream cells are referred to as I-cells. The observed I-cell is denoted as I^{O}-cell. In addition, any I-stream is modelled by a two-state IBP switching from ON to OFF and OFF to ON with probability $1-\alpha$ and $1-\beta$ per slot, respectively. That is, the mean time duration of an I-stream being in the ON and OFF states are $1/(1-\alpha)$ and $1/(1-\beta)$, respectively. Also, each I-stream generates λ cells/slot time in the ON state and generates no cell in the OFF state. Let i^n be the number of I-streams being in the ON state in the nth slot time, and B_{i^n} the number of I-cells arriving in the nth slot time given i^n I-streams in the ON state. The probability mass function $b_{i^n}(j)$ of B_{i^n} thus becomes

$$b_{i^n}(j)=\mathrm{Prob}[B_{i^n}=j]=\binom{i^n}{j}\cdot\lambda^j\cdot(1-\lambda)^{i^n-j},$$

$$0\le j\le i^n,\ 0\le i^n\le N_b.$$

Consequently, the transition probability that the number of I-streams in the ON state changes from i^{n-1} to i^n, p_{i^{n-1},i^n}, can be given as

$$p_{i^{n-1},i^n}=\sum_{i=0}^{i^{n-1}}\binom{i^{n-1}}{i}\alpha^i(1-\alpha)^{i^{n-1}-i}\binom{N_b-i^{n-1}}{i^n-i}$$
$$(1-\beta)^{i^{n-1}-i}\cdot\beta^{N_b-i^{n-1}-(i^n-i)},\quad 0\le i\le i^{n-1},$$
$$0\le i^n-i\le N_b-i^{n-1},\qquad(1)$$

where i is the number of I-streams still staying in the ON state and i^n-i is the number of I-streams switching from the OFF to the ON state. The steady state probability of j I-streams in the ON state, denoted as $\phi(j)$, can be directly computed by

$$\phi(j)=\sum_{i=0}^{N_b}\phi(i)\cdot p_{i,j},\quad 0\le j\le N_b,\qquad(2)$$

where $p_{i,j}$ is the transition probability defined in Eq. (1).

Moreover, an ATM switch is assumed to employ the output buffer (buffer size = K) mechanism and the FCFS (first come first served) service discipline. Simultaneously-arriving cells are served on a random basis. Each output buffer of a switch thus becomes a discrete-time single-server buffer-size-K queueing system, namely $M^{[N_1]}+I^{[N_2]}/D/1/K$. During the operation of the system, three events occur at the beginning and end of each slot time, as shown in Fig. 1. In Event 1, the number of I-streams in the ON state is changed from i^{n-1} to i^n. In Event 2, new cells arrive and are queued in the buffer. Finally, during Event 3, a cell (if any) departs and the first cell in the queue begins to be served.

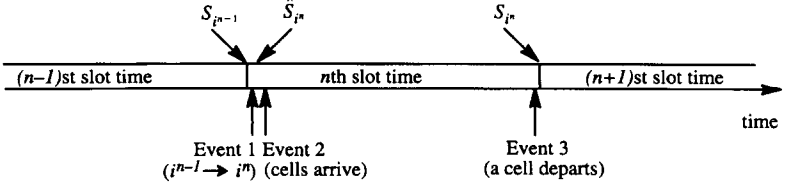


Fig. 1. Three events and system lengths.

In what follows, we first derive the system length distribution of the $M^{[N_1]}+I^{[N_2]}/D/1/K$ system. Based on the system length distribution, we then compute two performance metrics (the CD and CLR) which serve as the training data of the neural network presented below.

2.1. System length distribution

The system length distribution is examined at each slot time. Let $\hat{S}_{i^n}$ and S_{i^n} be the system lengths (with i^n I-streams in the ON state) observed at the nth slot time after the occurrence of Events 1 and 3, respectively. After Event 1 has occurred, the number of I-streams in the ON state is changed from i^{n-1} to i^n but the system length remains the same. Accordingly,

$$\hat{S}_{i^n} = S_{i^{n-1}}, \quad 0 \le i^{n-1} \le N_b,\ 0 \le i^n \le N_b. \tag{3}$$

After Event 2 occurs, the system length is incremented by the number of newly-arriving M-cells and I-cells but only up to the maximum system length, $K+1$. Moreover, owing to the departure of a cell after the occurrence of Event 3, the system length is decremented by 1 until zero. Thus,

$$S_{i^n} = \mathrm{MAX}\big(\mathrm{MIN}\big(\hat{S}_{i^n} + \Omega + B_{i^n}, K+1\big) - 1, 0\big),\ 0 \le i^n \le N_b, \tag{4}$$

where MAX and MIN are the maximum and minimum functions, respectively.

Let $\hat{s}_{i^n}(j)$ and $s_{i^n}(j)$ be the probability mass functions of $\hat{S}_{i^n}$ and S_{i^n}, respectively. From Eq. (3), $\hat{s}_{i^n}(j)$ is related to $s_{i^{n-1}}(j)$, $0 \le i^{n-1} \le N_b$, by

$$\hat{s}_{i^n}(j) = \sum_{i^{n-1}=0}^{N_b} p_{i^{n-1},i^n} \cdot s_{i^{n-1}}(j), \quad 0 \le i^n \le N_b,\ 0 \le j \le K, \tag{5}$$

where p_{i^{n-1},i^n} is the transition probability defined in Eq. (1). From Eq. (4), since the probability mass function of the sum of two independent random variables is the convolution of the individual probability mass functions, $s_{i^n}(j)$ can be given as

$$s_{i^n}(j) = \pi_1\big(\pi^{K+1}\big(\hat{s}_{i^n}(j+1) * m(j+1) * b_{i^n}(j+1)\big)\big),$$
$$0 \le i^n \le N_b,\ 0 \le j \le K, \tag{6}$$

where * is a convolution operator, and π_1 and π^{k+1} are the MAX and MIN functions, respectively, defined as

$$\pi_1(f(j)) = \begin{cases} 0 & j < 1 \\ f(0)+f(1) & j = 1 \\ f(j) & j > 1 \end{cases} \quad \text{and} \quad \pi^{K+1}(f(j)) =$$

$$\begin{cases} f(j) & j < K+1 \\ \displaystyle\sum_{i=K+1}^{\infty} f(i) & j = K+1 \\ 0 & j > K+1 \end{cases}$$

As a result, from Eq. (5) and Eq. (6), we can obtain $s_i(j)$, the limiting distribution of $s_{i^n}(j)$, by

$$s_i(j) = \lim_{n\to\infty} s_{i^n}(j), \quad 0 \le i \le N_b,\ 0 \le j \le K, \tag{7}$$

with initial condition

$$\sum_{i^0=0}^{N_b} \sum_{j=0}^{K} s_{i^0}(j) = 1.$$

2.2. CD and CLR

Having derived system length distribution $s_i(j)$, we are now at the stage of computing three performance metrics, namely the system time distribution, CD and CLR, for M-cells and I-cells.

2.2.1. M-cells

Let $\Omega_{\bar{0}}$ denote a positive number of M-cells arriving in a slot time. Thus, the probability mass function $m_{\bar{0}}(j)$, $(j>0)$ of $\Omega_{\bar{0}}$ becomes $m_{\bar{0}}(j) = m(j)/(1-m(0))$, $1 \le j \le N_a$. Furthermore, let $\tilde{m}_{\bar{0}}(j)$ be the probability mass function of a positive number of M-cells including the M^{O}-cell arriving in a slot time. From the renewal theory [23], $\tilde{m}_{\bar{0}}(j)$ is given as $\tilde{m}_{\bar{0}}(j) = j \cdot m_{\bar{0}}(j)/E[\Omega_{\bar{0}}]$, $1 \le j \le N_a$. Thus, with the M^{O}-cell included, the probability of a total number of h M-cells and I-cells, given i I-streams in the ON state, arriving in a slot time becomes $\{\tilde{m}_{\bar{0}}(h) * b_i(h)\}$. Owing to the fact that the probability of the M^{O}-cell being served jth among h cells is $1/h$, the probability mass function $r_{M,i}(j)$ of the M^{O}-cell being served jth among h cells becomes

$$r_{M,i}(j) = \sum_{h=j}^{N_a+N_b} \frac{\{\tilde{m}_{\bar{0}}(h) * b_i(h)\}}{h},$$
$$0 < N_a,\ 0 \le i \le N_b,\ 1 \le j \le N_a + N_b. \tag{8}$$

Now, notice that the system time for the M^{O}-cell is the sum of the time serving simultaneously-arriving but served-beforehand cells, and the time serving those cells already in the queue. The former term has just been derived in Eq. (8). The latter term is now derived. Due to the memoryless property of M-streams, the system length distribution possessed by the M^{O}-cell is thus identical to the general system length distribution, namely $s_i(j)$, which was previously derived in Eq. (7). Again, since the probability mass function of the sum of two independent random variables is the convolution of the individual probability mass function, the system time distribution $s_M(j)$ for M-cells is given as

$$s_M(j) = \sum_{i=0}^{N_b} s_i(j) * r_{M,i}(j), \quad 1 \le j \le K + N_a + N_b. \tag{9}$$

On account of the maximum system length of $K+1$ for M-cells, the CLR for M-cells (L_M) is acquired as

$$L_M = \sum_{j=K+2}^{K+N_a+N_b} s_i(j) * r_{M,i}(j). \tag{10}$$

Finally, the CD for M-cells (D_M) can be simply expressed as

$$D_M = \sum_{j=1}^{K+1} \frac{j \cdot s_M(j)}{(1 - L_M)}, \tag{11}$$

where $s_M(j)$ is normalized by $1 - L_M$, namely the probability of successfully-delivered M-cells.

2.2.2. I-cells

The system time for the I^O-cell is also determined by the sum of the time serving simultaneously-arriving but served-beforehand cells, and the time serving those cells already in the queue. The former term can be similarly derived as

$$r_{I,i}(j) = \sum_{h=j}^{N_a+N_b} \frac{\{m(h) * \tilde{b}_{i,\tilde{0}}(h)\}}{h}, \quad 0 < N_b,\ 0 \le i \le N_b,$$
$$1 \le j \le N_a + N_b. \tag{12}$$

However, due to the inapplicability of the memoryless property to I-streams, the system length distribution possessed by the I^O-cell, denoted as $\tilde{s}_i(j)$, is no longer the same as the general system length distribution $s_i(j)$ given in Eq. (7). To derive $\tilde{s}_i(j)$, let $\Phi_{\tilde{0}}$ be a positive number of I-streams in the ON state, and $\phi_{\tilde{0}}(i)$ $(i > 0)$ be the probability mass function of $\Phi_{\tilde{0}}$. Apparently, $\phi_{\tilde{0}}(i) = \phi(i)/(1 - \phi(0))$, $1 \le i \le N_b$. Further, let $\tilde{\phi}_{\tilde{0}}(i)$ be the probability mass function of i I-streams (in which the source of I^O-cell is included) being in the ON state. Again, from the renewal theory [23], $\tilde{\phi}_{\tilde{0}}(i)$ is obtained as $\tilde{\phi}_{\tilde{0}}(i) = i \cdot \phi_{\tilde{0}}(i)/E[\Phi_{\tilde{0}}]$, $1 \le i \le N_b$. Note that $\tilde{s}_i(j)$ is observed upon the arrivals of I-cells, and $s_i(j)$ is observed at each slot time. That is, $\sum_{j=0}^{K} \tilde{s}_i(j) = \tilde{\phi}_{\tilde{0}}(i)$ and $\sum_{j=0}^{K} s_i(j) = \phi(i)$. Owing to the fact that, under $j = a$ and $i = b$, the ratio of $\tilde{s}_b(a)$ to $s_b(a)$ is equal to $\tilde{\phi}_{\tilde{0}}(b)/\phi(b)$, $\tilde{s}_i(j)$ becomes

$$\tilde{s}_i(j) = s_i(j) \cdot \tilde{\phi}_{\tilde{0}}(i)/\phi(i), \quad 1 \le i \le N_b,\ 0 \le j \le K. \tag{13}$$

Hence, on the basis of Eq. (12) and Eq. (13), the system time distribution $s_I(j)$ for I-cells is given as

$$s_I(j) = \sum_{i=0}^{N_b} \tilde{s}_i(j) * r_{I,i}(j), \quad 1 \le j \le K + N_a + N_b. \tag{14}$$

Furthermore, on account of the maximum system length of $K + 1$ for I-cells, the CLR for I-cells (L_I) is acquired as

$$L_I = \sum_{j=K+2}^{K+N_a+N_b} \tilde{s}_i(j) * r_{I,i}(j). \tag{15}$$

Finally, the CD for I-cells (D_I) can be given by

$$D_I = \sum_{j=1}^{K+1} \frac{j \cdot s_I(j)}{(1 - L_I)} \tag{16}$$

where $s_i(j)$ is normalized by $1 - L_I$, namely the probability of successfully-delivered I-cells.

2.3. Confirmation of analytic results

To verify the accuracy of the analysis, we derived analytic results using MATLAB [24], and implemented the time-based simulation in the C language. Both the analytic computation and simulation were performed on a 586 PC with Intel Pentium-60 CPU. The program for analytic computation terminated when all entries of matrix $|s_{i^n}(j) - s_{i^{n-1}}(j)|$ dropped to 10^{-6} and below. The simulation program terminated when a loss of 10^4 cells had been detected. The traffic classes and their corresponding parameters used for both the analytic computation and simulation are summarized in Table 1. Fig. 2 and Fig. 3 depict the CD and CLR for $M_1\&I_1$ and $M_2\&I_3$ systems, respectively. Both figures demonstrate the profound agreement of analytic results with simulation results.

Fig. 2 shows the CD and CLR of each indicated traffic class as the number of I_1-streams increases while retaining aggregate loads (ρ) of 0.7 and 0.8. Notice that the aggregate load is defined as the total traffic load from M-streams and I-streams. For example, under an aggregate load of 0.8 in Fig. 2, an increase in the number of I_1-streams from 3 (0.1 × 3) to 4 (0.1 × 4) results in a decrease in the number of M_1-streams from 50 (0.01 × 50) to 40 (0.01 × 40). In addition, the figure shows that both the CD and CLR of each traffic class increase with the number of I_1-streams. This is because, under the same aggregate load, an increase in the number of I-streams (i.e., more high-burstiness traffic) results in a decrease in statistical multiplexing gain [4]. Fig. 3 displays the CD and CLR of each marked traffic class as a function of the aggregate load. The figure shows that both the CD and CLR increase with the aggregate load. Moreover, the figure also exhibits that the larger the number of M_2-streams (low-burstiness), the lower the CD and CLR.

3. NNCAC mechanism

3.1. Principles

The NNCAC mechanism has been designed based on two

Table 1
Traffic classes and parameters

Class	Parameter
M_1	$R_1 = 0.01$
M_2	$R_2 = 0.05$
M_3	$R_3 = 0.025$
I_1	$ON_1 = 5$; $OFF_1 = 45$; $\lambda_1 = 1.0$
I_2	$ON_2 = 5$; $OFF_2 = 95$; $\lambda_2 = 1.0$
I_3	$ON_3 = 7.5$; $OFF_3 = 67.5$; $\lambda_3 = 1.0$
I_4	$ON_4 = 2.5$; $OFF_4 = 97.5$; $\lambda_4 = 1.0$
I_5	$ON_5 = 10$; $OFF_5 = 190$; $\lambda_5 = 1.0$

R_x: Mean cell arrival rate (cell/slot time) for a class-x stream.
ON_x: Mean ON length of an I_x-stream.
OFF_x: Mean OFF length of an I_x-stream.
λ_x: Mean cell arrival rate of an I_x-stream in the ON state.

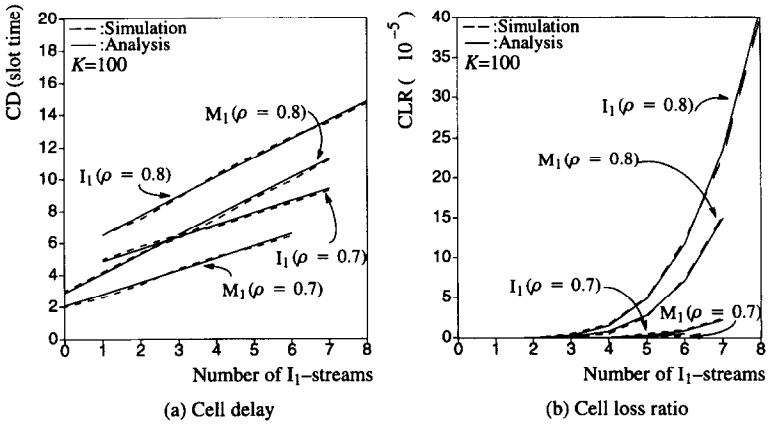


Fig. 2. CD and CLR under two aggregate loads.

principles. First, notice that to make a call acceptance decision in real time the CAC mechanism requires a massive amount of data to be determined and saved in advance. Rather than directly save these data in storage, our NNCAC mechanism employs a neural network method which has been widely accepted as an effective method [17,18,25] of on-line decision making for ATM traffic control. A brief overview of the neural network method will be described in the next subsection.

The second design principle is that, instead of using simulation results as the off-line training data, we employ analytic results (obtained from the previous section) for the training of the neural network. The rationale behind this is that, despite the fact that the amount of training time is disregarded owing to the off-line nature of the training, the determination of whether the simulation has converged to steady state is non-trivial under low-loss conditions of networks. This is further justified in the following context.

To compare the execution time of analytic computation and simulation until outputs converge to steady state, we performed an experiment on three desired ranges of CLRs, namely 10^{-4}, 10^{-5} and 10^{-7}, under three different numbers of I_1 and M_2 streams. Experimental results are shown in Fig. 4 and summarized in Table 2. In Fig. 4, (a), (c) and (e) respectively depict three ranges of CLRs within a shorter time duration, 4000 seconds, whereas (b), (d) and (f) show CLRs within a much longer time duration, for instance, 10^4 of 10^5 seconds. The figure clearly shows that the time to convergence via analytic computation is totally irrelevant to the CLR but is related to the total number of traffic streams. An increase in the number of streams yields a rise in time to convergence. On the contrary, the time to convergence via simulation is largely dependent on the CLR. In particular, the smaller the CLR the longer the time to convergence. This is owing to the fact that more samples are required to reach the steady state should the loss probability be smaller. For instance, we have observed that, to collected 10 000 cell losses in simulation, one has to perform the simulation for time periods of 1400, 47 000 and 2 300 000 seconds in the cases of CLRs of 10^{-4}, 10^{-5} and 10^{-7}, respectively.

3.2. Neural network training method—an overview

In the NNCAC mechanism, we employ the backpropagation learning method [26] for the off-line training of the neural network. The method basically approximates an arbitrary nonlinear function y being equal to $f(x)$ by adjusting

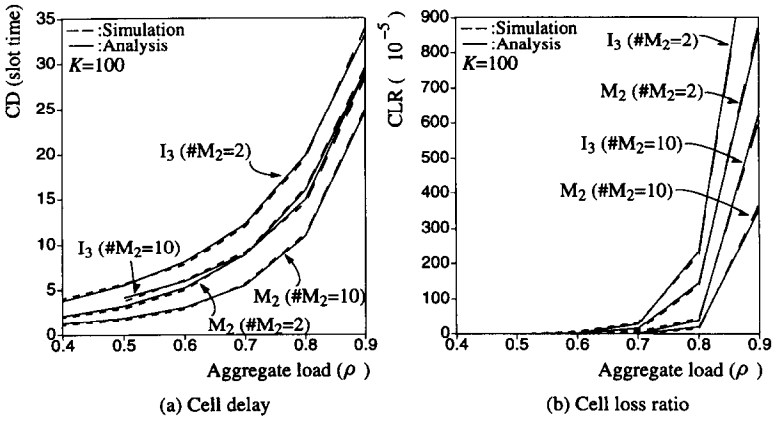
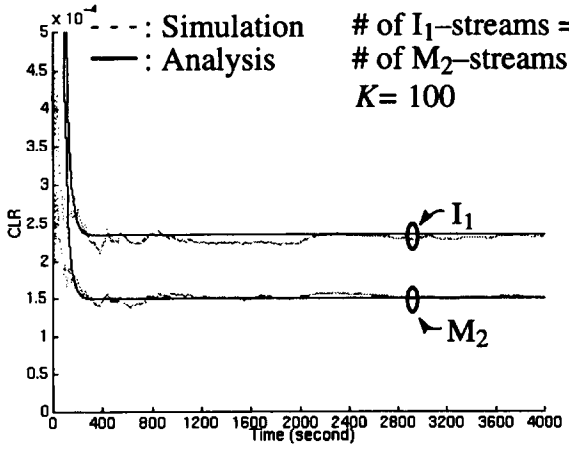
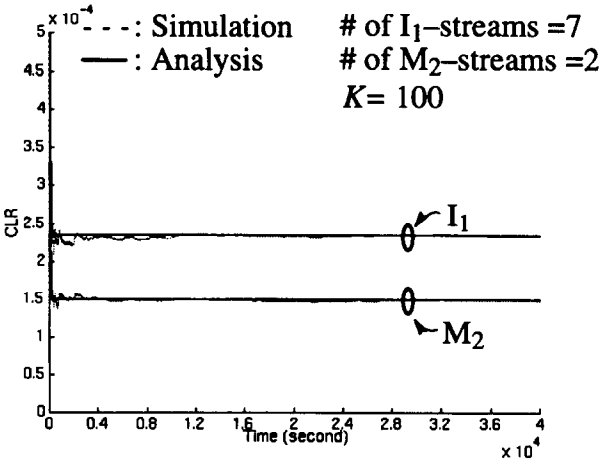


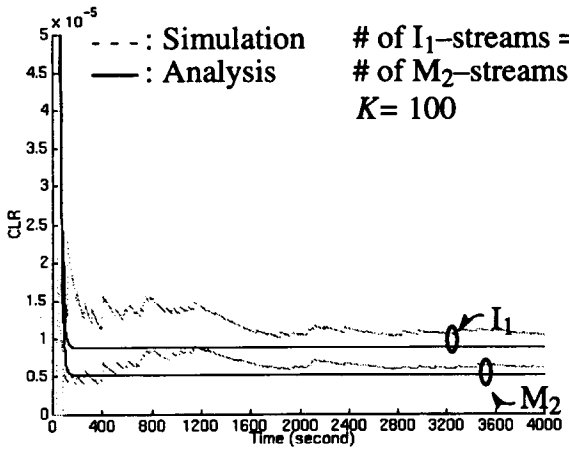
Fig. 3. CD and CLR as a function of load.



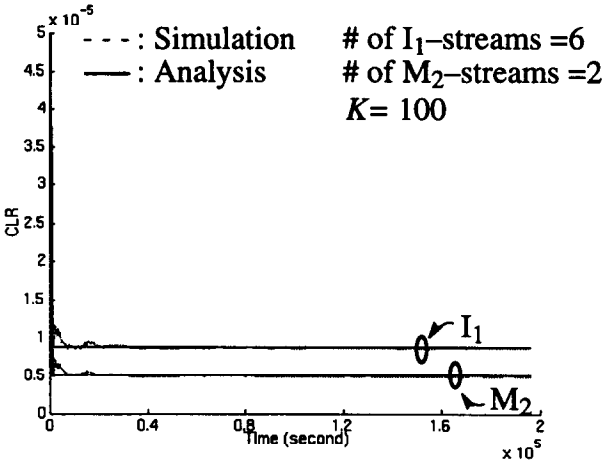
(a) CLR variation in short time period (CLR~10^{-4}).



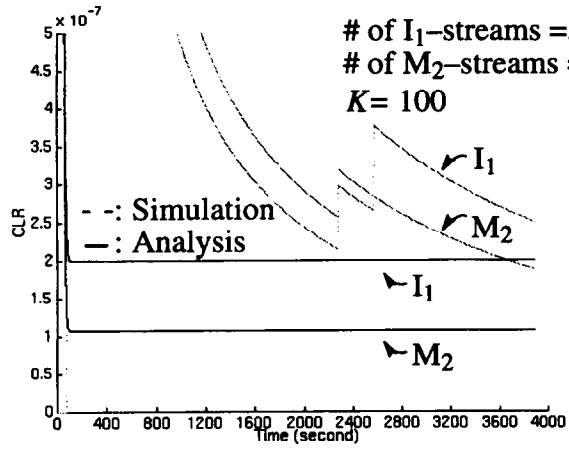
(b) CLR variation in long time period (CLR~10^{-4}).



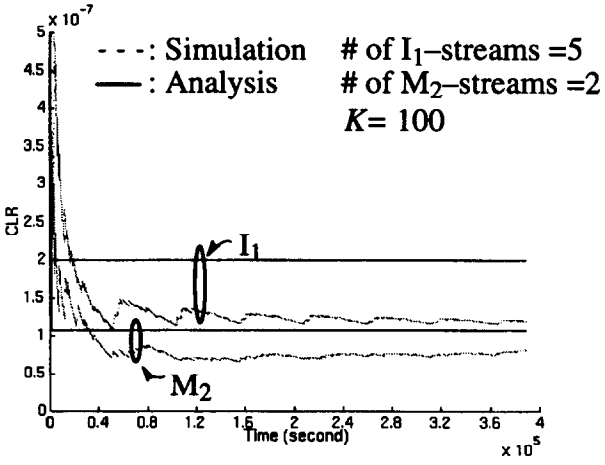
(c) CLR variation in short time period (CLR~10^{-5}).



(d) CLR variation in long time period (CLR~10^{-5}).



(e) CLR variation in short time period (CLR~10^{-7}).



(f) CLR variation in long time period (CLR~10^{-7}).

Fig. 4. Comparisons of time to convergence between analytic computation and simulation.

Table 2
Summary of the time to convergence

CLR	10^{-4}	10^{-5}	10^{-7}
Analysis	320 seconds	180 seconds	140 seconds
Simulation	11000 seconds	20000 seconds	> 400000 seconds

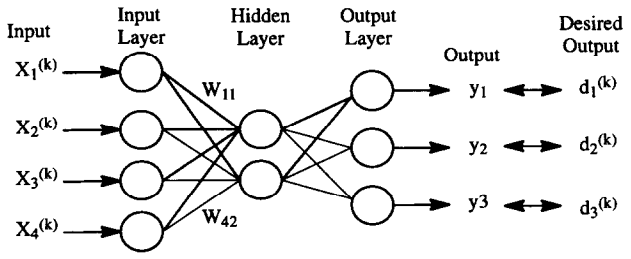


Fig. 5. Backpropagation neural network.

the weights in a backpropagation network according to all sampled (x,y) pairs.

Fig. 5 shows an example of a backpropagation neural network with three layers: input layer, hidden layer, and output layer. Each layer has a number of processing elements (or neurons) which are fully interconnected via adaptive weights (w_{ij}), by which the output of the ith neuron is connected with the input to the jth neuron. The neurons on the input layer simply store input data. These data are in turn manipulated by hidden and output layer neurons to perform two subsequent operations—weighted summation and normalization, as given in Eq. (17) and Eq. (18), respectively:

$$V_j = \sum_{i=1}^{n} W_{ij} X_i \tag{17}$$

$$O_j = \frac{1}{1 + \exp(-V_j)} \tag{18}$$

The backpropagation network learns by making changes in its weights in direction to minimize a given error function ($E(X)$) between its predictions and the training data set. That is,

$$w_i(t+1) = w_i(t) - c\frac{\partial E(X)}{\partial w_i} \tag{19}$$

where $w_i(t)$ is one of weights at cycle t, and c is the learning constant [26].

3.3. NNCAC system architecture

The NNCAC mechanism is composed of two phases: the off-line training phase and the on-line operation phase. As shown in Fig. 6, in the off-line training phase, the number of calls in each class (inputs) and the analytic results of CDs and CLRs (desired outputs) are all normalized to the range 0 to 1. All the weights are then learned at the end of the training phase. Based on the determined weights, during the on-line operation phase, the NNCAC mechanism then determines if a newly-arriving call of a class can be accepted by means of comparing the design CD or CLR output with the QOS threshold for such a class.

3.4. Experimental results

To demonstrate the viability of the NNCAC mechanism, we drew comparisons of the CD and CLR, in Fig. 7 and Fig. 8 respectively, between NNCAC results and analytic results under four cases. The call arrangement in each case is summarized as follows. First of all, for case 1, given 1 call of class I_2, 8 calls of class M_2, and 2 calls of class I_4, the number of calls of class I_5 alters from 3 to 8, resulting in an aggregate load of 0.65 to 0.90. In case 2, given 1 call of class I_2, 4 calls of class I_4, and 8 calls of class I_5, the number of calls of class M_2 alters from 2 to 7, resulting in the same aggregate load of 0.65 to 0.90. As for case 3, given 8 calls of class M_2, and 2 calls of each class I_4 and I_5, the number of calls of class I_2 alters from 2 to 7; and for case 4, given 1 call of class I_2, 8 calls of class M_2, and 2 calls of class I_5, the number of calls of class I_4 alters from all even numbers between 4 and 14 inclusive.

As shown in Fig. 7 and Fig. 8, the CD and CLR increase

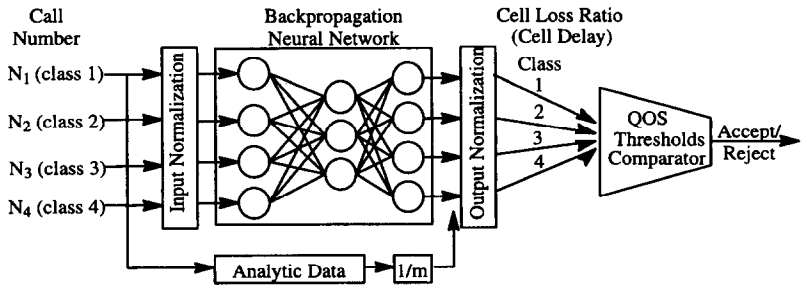


Fig. 6. NNCAC architecture.

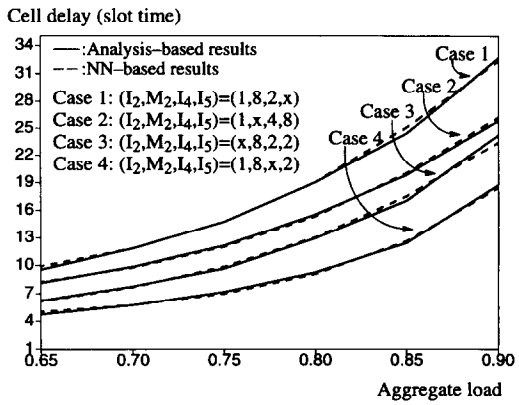


Fig. 7. CD as a function of aggregate load.

with the load of the network. In particular, traffic streams of high burstiness incur greater values of CD and CLR under any given aggregate load. For example, as shown in Fig. 7, case 1 traffic condition yields the highest CD owing to the dominance of the highest-burstiness call, I_5 in this case. Finally, most significantly, both figures show that the neural network method yields high precision of the CD and CLR compared with analytic results.

4. Conclusions

The paper has proposed a highly efficient and precise neural-network-based call admission control (NNCAC) mechanism in ATM networks. We initially provided an analysis of the cell delay and cell loss ratio for each traffic call based on a queueing system with dual heterogeneous arrivals (Bernoulli and IBP). The model has been expanded into a four-class system. The resulting four-class analytic data were then employed as the training data of the neural network during the off-line training phase. In the on-line operation phase, the pre-trained neural network then estimated both the cell delay and cell loss ratio of each traffic class and in turn determined the acceptance or rejection of a newly-arriving call. Finally, numerical results showed that our NNCAC results agreed with analytic results with negligible discrepancy.

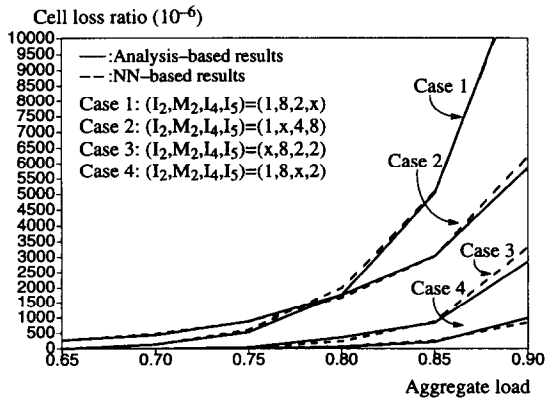


Fig. 8. CLR as a function of aggregate load.

References

[1] R. Händel, M.N. Huber, S. Schröder, ATM Networks—Concept, Protocol, Applications, 2nd ed., Addison-Wesley, 1993.
[2] M.D. Prycker, Asynchronous Transfer Mode Solution for Broadband ISDN, Ellis Horwood, 1993.
[3] ITU Study Group 13, Traffic Control and Congestion Control in B-ISDN, Draft Recommendation I.371, March 1994.
[4] H. Saito, Teletraffic Technologies in ATM Networks, Artech House, 1994.
[5] D. Yates, J. Kurose, D. Towsley, M.G. Hiuchyj, On pre-session end-to-end delay and the call admission problem for real-time applications with QOS requirements, ACM SIGCOMM'93, 1993, pp. 2–12.
[6] X. Chen, Modeling connection admission control, IEEE INFOCOM'93, 1993, pp. 274–281.
[7] K.T. Cho, S.K. Kawasaki, Call admission control method in ATM networks, IEEE ICC'92, 1992, pp. 1628–1633.
[8] K. Sohraby, Heavy traffic multiplexing behavior of highly-bursty heterogeneous sources and their admission control in high-speed networks, IEEE GLOBECOM' 92, 1992, pp. 1518–1523.
[9] A.I. Elwalid, D. Mitra, Effective bandwidth of general Markovian traffic sources and admission control of high speed networks, IEEE INFOCOM'93, 1993, pp. 256–265.
[10] H. Saito, K. Shiomoto, Dynamic call admission control in ATM networks, IEEE J. Select. Areas Commun. 9 (7) (1991) 982–989.
[11] T.H. Lee, K.C. Lai, S.T. Duann, Real time call admission control for ATM networks with heterogeneous bursty traffic, IEEE ICC'94, 1994, pp. 80–85.
[12] T. Murase, H. Suzuki, S. Sato, T. Takeuchi, A call admission control algorithm for ATM network using a simple quality estimate, IEEE J. Select. Areas Commun. 9 (9) (1991) 1461–1470.
[13] A. Dailianas, A. Bovopoulos, Real-time admission control algorithm with delay and loss guarantee in ATM networks, IEEE INFOCOM'94, 1994, pp. 1065–1072.
[14] F.Y.S. Lin, J.R. Yee, A real-time distributed routing and admission control algorithm for ATM networks, IEEE INFOCOM'93, 1993, pp. 792–801.
[15] R. Morris, B. Samadi, Neural network control of communications systems, IEEE Trans. Neural Networks 5 (4) (1994) 639–650.
[16] A. Hiramatsu, ATM communications network control by neural networks, IEEE Trans. Neural Networks 1 (1) (1990) 122–130.
[17] N. Ogino, Y. Wakahara, Application of neural network in ATM call admission control based on cell transfer state monitoring with dynamic threshold, IEICE Trans. Commun. E78-B (4) (1995) 465–475.
[18] B. Yuhas, Neural Networks in Telecommunications, Kluwer Academic, 1994.
[19] E.P. Rathgeb, Modeling and performance comparison of policing mechanisms for ATM networks, IEEE J. Select. Areas Commun. 9 (3) (1991) 325–334.
[20] Y. Ohba, M. Murata, H. Miyahara, Analysis of interdeparture processes for bursty traffic in ATM networks, IEEE J. Select. Areas Commun. 9 (3) (1991) 468–476.
[21] J.Y. Hui, Resource allocation for broadband networks, IEEE J. Select. Areas Commun. 6 (9) (1988) 1598–1608.
[22] P.L. Tien, J.M. Hah, M.C. Yuang, Neural-network-based call admission control for ATM networks with heterogeneous arrivals, Technical Report under Project CCL-G4-85012-10, National Chiao Tung University, Taiwan, 1996.
[23] J.N. Daigle, Queueing theory for telecommunications, Addison-Wesley, 1992.

[24] The Math Works Inc., MATLAB: High-Performance Numeric Computation and Visualization Software, 1992.
[25] A. Tarraf, I. Habib, T. Saadawi, Intelligent traffic control for ATM broadband networks, IEEE Communications Magazine 33 (10) (1995) 76–82.
[26] J. Zurada, Introduction to Artificial Neural Systems, West Publishing, 1992.

Jen M. Hah was born in Taiwan, ROC, in 1966. He received the B.S. degree in electrical engineering from the National Cheng Kung University, Taiwan, in 1989; the M.S. degree in computer science from the National Tsing Hua University, Taiwan, in 1991; and the Ph.D. degree in computer science and information engineering from the National Chiao Tung University, Taiwan, in 1997. His current research interests include high-speed networking, and performance modelling and analysis.

Po L. Tien was born in Taiwan, in 1969. He received the B.S. degree in applied mathematics, and the M.S. degrees in computer and information science, from the National Chiao Tung University, Taiwan, in 1992 and 1995, respectively. He is currently a Ph.D. candidate in the department of computer science and information engineering at the National Chiao Tung University, Taiwan. His current research interests include high-speed networking, multimedia communications, performance modelling and analysis, and applications of artificial neural networks.

Maria C. Yuang received the B.S. degree in applied mathematics from the National Chiao Tung University, Taiwan, in 1978; the M.S. degree in computer science from the University of Maryland, College Park, Maryland, in 1981; and the Ph.D degree in electrical engineering and computer science from the Polytechnic University, Brooklyn, New York, in 1989. From 1981 to 1990, she was with AT and T Bell Laboratories and Bell Communications Research (Bellcore), where she was a member of technical staff working on high-speed networking and protocol engineering. She has been an associate professor in computer science and information engineering at the National Chiao Tung University, Taiwan, since 1990. Her current research interests include high-speed networking, multimedia communications, performance modelling and analysis, and ATM network management.