Estimation-based call admission control with delay and loss guarantees in ATM networks

J.M. Hah
M.C. Yuang

Indexing terms: Call admission control (CAC), Quality of service, Asynchronous transfer mode, Cell delay, Cell loss ratio, Dual-arrival queueing model

Abstract: Call admission control (CAC) has been accepted as a potential solution for supporting a variety of traffic sources demanding different quality of service guarantees in asynchronous transfer mode networks. Basically, CAC is required to consume a minimum of time and space to make call acceptance decisions. In the paper a CAC algorithm is presented based on a novel estimation method, called quasilinear dual-class correlation (QLDC). All heterogeneous traffic calls are initially categorised into various classes. According to the number of calls in each traffic class, QLDC conservatively and precisely estimates the cell delay and cell loss ratio for each traffic class via simple vector multiplication. These vectors are computed in advance from the results of three dual arrival queuing models, $M^{[N_1]} + I^{[N_2]}/D/1/K$, $M_1^{[N_1]} + M_2^{[N_2]}/D/1/K$ and $I_1^{[N_1]} + I_2^{[N_2]}/D/1/K$, where M and I represent the Bernoulli process and the interrupted Bernoulli process, respectively. Consequently, the authors' QLDC-based CAC, as will be shown, yields low time complexity O(C) (in vector multiplications) and space complexity O(WC^2) (in bytes), where C is the total number of traffic classes and W is the total number of aggregate load levels. Numerical examples are also employed to justify that QLDC-based estimated results profoundly agree with simulation results in both the single-node and end-to-end cases.

1 Introduction

Asynchronous transfer mode (ATM) networks [1, 2] have been expected to fully utilise network resources while retaining satisfactory quality of service (QoS) for each user in broadband ISDNs [3]. To satisfy this requirement, call admission control (CAC) [4] has been 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 [6–13] and delay-and-loss-based [14, 15].

In the delay-based category, the mechanism [5] approximated (through simulation) the end-to-end delay distribution under a particular network model on which CAC was based. In the loss-based category, the CAC mechanism [6] established a simple computation procedure based on a quasistationary approximation for the solution of an MMPP/G/1/K queue. The CAC method [7] calculated the cell loss ratio (CLR) by means of the probability mass function (PMF) of the number of cells transferred from multiplexed calls and used recursive equations to reduce the amount of calculation. The CAC scheme [8] adopted the tail of the queue length distribution as a simple metric for highly bursty heavy traffic networks. The CAC scheme [9] assigned the bandwidth to each call subject to a small (~10^{-9}) CLR. The bandwidth was the maximal real eigenvalue of a matrix directly obtained from the source characteristics and the admission criteria. The CAC mechanism [10] adopted an estimated distribution of the number of cells arriving during a fixed interval to evaluate the upper bound of the CLR. The CAC methods [11, 12] assumed that the QoSs for all traffic classes are identical. The former method performed CAC by means of simple multiplication and division operations. The latter method modelled each traffic source as a Bernoulli process. Acceptance decision making was based on whether the current load exceeds a precalculated threshold chosen from Bayesian decision theory. The scheme [13] employed a method of estimating the CLR for each traffic class, and performed CAC based on both the virtual bandwidth and virtual link capacity concepts [16].

In the delay-and-loss-based category, the algorithm [14] was based on the time framing strategy to provide guarantees in terms of delay, delay jitter and the upper bounds of the burst and CLR. The algorithm [15] assumed that cell arrivals to each input port formed a simple Bernoulli process and formulated the CAC problem as a nonlinear combinatorial optimisation problem. Basically, 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, but at the expense of an increase in the time and space complexity.

In this paper we present a delay-and-loss-based CAC algorithm using a novel estimation method, called quasilinear dual class correlation (QLDC). All heterogeneous traffic calls are initially categorised into various

IEE Proceedings online no. 19970958

Paper first received 20th September 1995 and in revised form 2nd October 1996

The authors are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, Republic of China

classes. According to the number of calls in each traffic class, QLDC conservatively estimates the cell delay (CD) and CLR for each traffic class via simple vector multiplication. These vectors are computed in advance from the results of three dual arrival queuing models, $M^{[N_1]} + I^{[N_2]}/D/1/K$, $M_1{}^{[N_1]} + M_2{}^{[N_2]}/D/1/K$ and $I_1{}^{[N_1]} + I_2{}^{[N_2]}/D/1/K$, where M and I represent the Bernoulli process and the Interrupted Bernoulli Process (IBP), respectively. Consequently, our QLDC-based CAC, as will be shown, yields low time complexity $O(C)$ (in vector multiplications) and space complexity $O(WC^2)$ (in bytes), where C is the total number of traffic classes and W is the total number of aggregate load levels. We also employ numerical examples to justify that QLDC-based estimated results profoundly agree with simulation results in both single node and end-to-end cases.

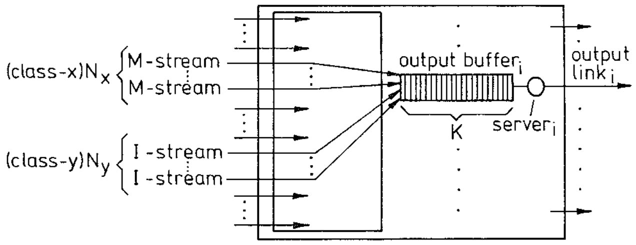


Fig. 1 *$M^{[N_x]} + I^{[N_y]}/D/1/K$ queuing system*

2 Queuing model and analysis

All traffic source streams (calls) are categorised into various classes (see Fig. 1) 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. For example, streams of file transfer can be classified as one class, and streams of video based on a particular compression method can be classified as another class. Moreover, any nonbursty stream (such as files, or any stream output from a traffic shaper [17]) is modelled by a Bernoulli process (called an M-stream), and any bursty stream (such as voice or video) is modelled by an IBP (called an I-stream). The combinational use of both processes has been widely accepted to model multiplexed traffic in ATM networks [18, 19].

Owing to the dual-class consideration by QLDC, we hereinafter examine the system with only two classes: two M-stream classes, two I-stream classes, or one M-stream class and one I-stream class. For simplicity of illustration, we focus on analysing the last system, namely the system with one M-stream class (referred to as class 1) and one I-stream class (referred to as class 2). For class 1, we further assume that there are N_1 M-streams. These M-stream cells are referred to as M-cells. The observed M-cell is denoted as the M°-cell. Let Ω be the number of M-cells arriving in a slot time, and R the mean cell arrival rate (cells/slot time). The PMF of Ω, denoted as $m(j)$, follows a binomial distribution, namely

$$m(j) = \text{prob}[\Omega = j] = \binom{N_1}{j} R^j (1-R)^{N_1-j}$$
$$0 \leq j \leq N_1$$

For class 2, we assume that there are N_2 I-streams. These I-stream cells are referred to as I-cells. The observed I-cell is denoted as the I°-cell. In one slot time, an I-stream changes from state ON to OFF with probability $1-\alpha$ and from state OFF to ON with probability $1-\beta$ 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. Besides, 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 in the ON state at the nth slot time, and B_{i^n} the number of I-cells arriving at the nth slot time given i^n I-streams in the ON state. The PMF of B_{i^n}, denoted as $b_{i^n}(j)$, follows a binomial distribution, namely

$$b_{i^n}(j) = \text{prob}[B_{i^n} = j] = \binom{i^n}{j} \lambda^j (1-\lambda)^{i^n - j}$$
$$0 \leq j \leq i^n, \quad 0 \leq i^n \leq N_2$$

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

$$p_{i^{n-1}i^n} = \sum_{i=0}^{i^{n-1}} \left\{ \binom{i^{n-1}}{i} \alpha^i (1-\alpha)^{i^{n-1}-i} \binom{N_2 - i^{n-1}}{i^n - i} \times (1-\beta)^{i^n - i} \beta^{N_2 - i^{n-1} - (i^n - i)} \right\} \quad (1)$$

The steady-state probability of j I-streams in the ON state, denoted as $\phi(j)$, can be computed by

$$\phi(j) = \sum_{i=0}^{N_2} (\phi(i) \cdot p_{ij}), \quad 0 \leq j \leq N_2 \quad (2)$$

where p_{ij} is the transition probability defined in eqn. 1.

Moreover, an ATM switch is assumed to employ the output buffering (buffer size = K) mechanism and the FCFS 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 queuing 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. 2. 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 departs and the first cell in the queue begins to be served.

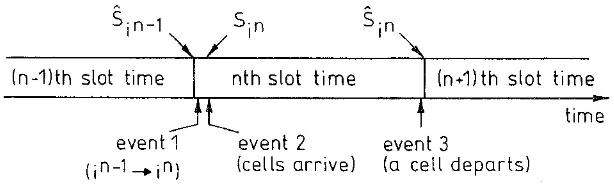


Fig. 2 *Events and system length*

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 three performance metrics (the system time distribution, CD and CLR) which serve as the base of the CAC algorithm presented afterwards.

2.1 System length distribution

The system length distribution is first examined at each slot time. Let S_{i^n} and $\hat{S}_{i^n}$ be the system lengths given i^n I-streams in the ON state after the occurrence of events 1 and 3 observed at the nth slot time, respectively. Accordingly,

$$S_{i^n} = \hat{S}_{i^{n-1}}, \quad 0 \leq i^{n-1} \leq N_2, \quad 0 \leq i^n \leq N_2 \quad (3)$$

$$\hat{S}_{i^n} = \max\left(\min\left(S_{i^n} + \Omega + B_{i^n}, K+1\right) - 1, 0\right)$$
$$0 \le i^n \le N_2 \tag{4}$$

Let $s_{i^n}(j)$ and $\hat{s}_{i^n}(j)$ be the PMFs of S_{i^n} and $\hat{S}_{i^n}$, respectively. From eqn. 3, $s_{i^n}(j)$ becomes

$$s_{i^n}(j) = \sum_{i^{n-1}=0}^{N_2} \left(p_{i^{n-1}i^n} \cdot \hat{s}_{i^{n-1}}(j)\right)$$
$$0 \le i^n \le N_2, \quad 0 \le j \le K \tag{5}$$

where $p_{i^{n-1}i^n}$ is defined in eqn. 1. From eqn. 4, $\hat{s}_{i^n}(j)$ can be given as

$$\hat{s}_{i^n}(j) = \pi_1\left(\pi^{K+1}\left(s_{i^n}(j+1) * m(j+1) * b_{i^n}(j+1)\right)\right)$$
$$0 \le i^n \le N_2, \quad 0 \le j \le K \tag{6}$$

where * is a convolution operator, and π_1 and π^{K+1} are the *max* and *min* functions, respectively, defined as

$$\pi_1(f(j)) = \begin{cases} 0 & j < 1 \\ f(0) + f(1) & j = 1 \\ f(j) & j > 1 \end{cases}$$

and

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

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

$$s_i(j) = \lim_{n\to\infty} s_{i^n}(j) \quad 0 \le i \le N_2, \quad 0 \le j \le K \tag{7}$$

with initial condition $\Sigma_{i^0=0}^{N_2} \Sigma_{j=0}^{K} \hat{s}_{i^0}(j) = 1$.

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 the positive number of M-cells arriving in a slot time, and $m_{\bar{0}}(j)$ be its PMF. $m_{\bar{0}}(j)$ is given as $m_{\bar{0}}(j) = m(j)/(1-m(0))$, $1 \le j \le N_1$. Furthermore, let $\tilde{m}_{\bar{0}}(j)$ be the PMF of the number of M-cells including the M°-cell arriving in a slot time. From renewal theory [20], $\tilde{m}_{\bar{0}}(j)$ is obtained as $\tilde{m}_{\bar{0}}(j) = jm_{\bar{0}}(j)/E[\Omega_{\bar{0}}]$, $1 \le j \le N_1$, where E is the mean function. Thus, with the M°-cell included, the probability of a total number of h M-cells and I-cells arriving in a slot time given i I-streams in the ON state becomes $\{\tilde{m}_{\bar{0}}(h) * b_i(h)\}$. Owing to the fact that the probability of the M°-cell being served jth among h cells is $1/h$, the probability $r_{M,i}(j)$ for the M°-cell being served jth among simultaneously arriving cells becomes

$$r_{M,i}(j) = \sum_{h=j}^{N_1+N_2} \frac{\tilde{m}_{\bar{0}}(h) * b_i(h)}{h}$$
$$0 < N_1, \quad 0 \le i \le N_2, \quad 1 \le j \le N_1 + N_2 \tag{8}$$

Now, notice that the system time for the M°-cell is the sum of the order by which the M°-cell is served among simultaneously arriving cells and the number of cells already in the queue. The former term has just been derived in eqn. 8. The latter term is now derived. Owing to the memoryless property of M-streams, the system length distribution possessed by M-streams is thus identical to the general system length distribution $s_i(j)$ derived in eqn. 7. Hence, the system time distribution $s_M(j)$ for M-cells is given as

$$s_M(j) = \sum_{i=0}^{N_2} \left(s_i(j) * r_{M,i}(j)\right), \quad 1 \le j \le K+1 \tag{9}$$

As a result, the CLR for M-cells (L_M) is acquired as

$$L_M = \sum_{j=K+2}^{K+N_1+N_2} \left(s_i(j) * r_{M,i}(j)\right) \tag{10}$$

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

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

2.2.2 I-cells: The system time for the I°-cell is also the sum of the order by which the I°-cell is served among simultaneously arriving cells and the number of cells already in the queue. The former term can be derived similarly as

$$r_{I,i}(j) = \sum_{h=j}^{N_1+N_2} \frac{m(h) * \tilde{b}_{i,\bar{0}}(h)}{h}$$
$$0 < N_2, \quad 0 < i \le N_2, \quad 1 \le j \le N_1 + N_2 \tag{12}$$

However, due to the inapplicability of the memoryless property to I-streams, the system length distribution possessed by I-streams, denoted as $\tilde{s}_i(j)$, is different from the general system length distribution $s_i(j)$ given in eqn. 7. To derive $\tilde{s}_i(j)$, let $\Phi_{\bar{0}}$ denote the positive number of I-streams in the ON state and $\phi_{\bar{0}}(i)$ be its PMF. $\phi_{\bar{0}}(i)$ is given as $\phi_{\bar{0}}(i) = \phi(i)/(1-\phi(0))$, $1 \le i \le N_2$. Further, let $\tilde{\phi}_{\bar{0}}(i)$ be the PMF of i I-streams (in which the source of the I°-cell is included) in the ON state. Again, from renewal theory, $\tilde{\phi}_{\bar{0}}(i) = i\phi_{\bar{0}}(i)/E[\Phi_{\bar{0}}]$, $1 \le i \le N_2$. Note that $\tilde{s}_i(j)$ is examined on arrivals of I-cells, and $s_i(j)$ is examined at each slot time. That is, $\Sigma_{j=0}^{K} \tilde{s}_i(j) = \tilde{\phi}_{\bar{0}}(i)$ and $\Sigma_{j=0}^{K} s_i(j) = \phi(i)$. Owing to the fact that the ratio of $\tilde{s}_i(j)$ to $s_i(j)$ is equal to $\tilde{\phi}_{\bar{0}}(i)/\phi(i)$, $\tilde{s}_i(j)$ becomes

$$\tilde{s}_i(j) = s_i(j)\tilde{\phi}_{\bar{0}}(i)/\phi(i) \quad 1 \le i \le N_2, \quad 0 \le j \le K \tag{13}$$

Hence, from eqns, 12 and 13, the system time distribution $s_I(j)$ for I-cells is given as

$$s_I(j) = \sum_{i=0}^{N_2} \left(\tilde{s}_i(j) * r_{I,i}(j)\right), \quad 1 \le j \le K+1 \tag{14}$$

As a result, the CLR for I-cells (L_I) is acquired as

$$L_I = \sum_{j=K+2}^{K+N_1+N_2} \left(\tilde{s}_i(j) * r_{I,i}(j)\right) \tag{15}$$

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

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

3 Experimental results

To verify the accuracy of the analysis, we derived analytical results using MATLAB [21], and implemented the time-based simulation in the C language. Characteristics of traffic classes used in this and the following Sections are summarised in Table 1. Figs. 3–5 depict the CD and CLR for M_1&I_1, M_2&I_2 and I_3&I_4 systems, respectively. All Figures demonstrate the profound agreement of the analytical results with the simulation results.

Table 1: Characteristics of traffic classes

Traffic class	Traffic parameter		
M_1	$R_1 = 0.05$		
M_2	$R_2 = 0.01$		
I_1	$ON_1 = 5;$	$OFF_1 = 45;$	$\lambda_1 = 1.0$
I_2	$ON_2 = 20;$	$OFF_2 = 180;$	$\lambda_2 = 1.0$
I_3	$ON_3 = 2.5$	$OFF_3 = 47.5;$	$\lambda_3 = 1.0$
I_4	$ON_4 = 10;$	$OFF_4 = 90;$	$\lambda_4 = 1.0$

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

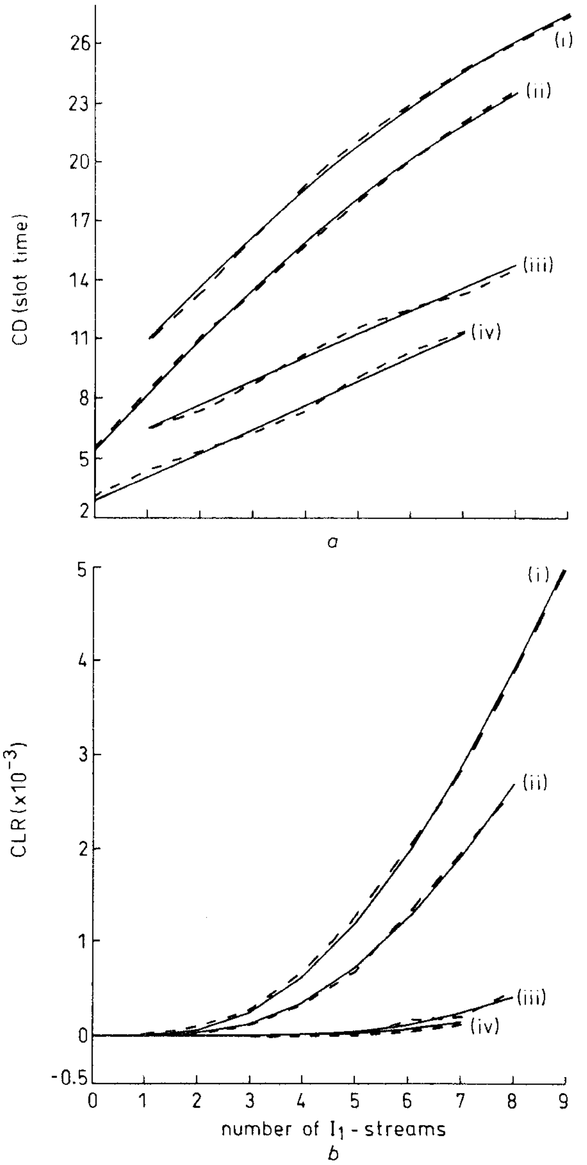
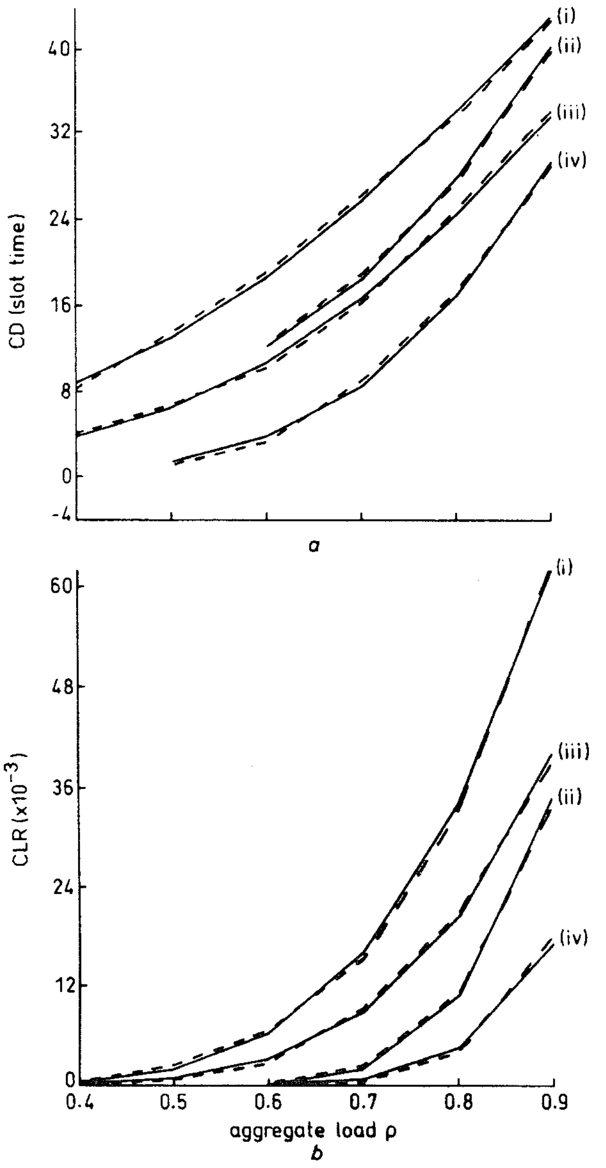


Fig. 3 *CD and CLR as functions of the number of I_1-streams*
a CD
b CLR
(i) I_1 (ρ = 0.9); (ii) M_1 (ρ = 0.9); (iii) I_1 (ρ = 0.8); (iv) M_1 (ρ = 0.8)
K = 100
– – – – simulation
——— analysis

Fig. 3 shows the CD and CLR of each indicated traffic class as the number of I_1-streams increases while retaining aggregate loads ρ of 0.8 and 0.9 under a buffer size K of 100. Note that the aggregate load is defined as the total traffic load from the M-streams and I-streams. For example, under an aggregate load of 0.8 in Fig. 3, an increase in the number of I_1-streams from three (0.1 × 3) to four (0.1 × 4) results in a decrease in the number of M_1-streams from ten (0.05 × 10) to eight (0.05 × 8). 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].



Fi
a CD
b CLR
(i) I_2 (10 M_2 streams); (ii) I_2 (50 M_2 streams); (iii) M_2 (10 M_2 streams); (iv) M_2 (50 M_2 streams)
K = 100
– – – – simulation
——— analysis

Fig. 4 displays the CD and CLR of each marked traffic class as functions of the aggregate load. The Figure shows that both the CD and CLR increase with the aggregate load under numbers of M_2-streams (low burstiness) of ten and 50. Moreover, the Figure also exhibits that the larger the number of M_2-streams the lower are CD and CLR. Fig. 5 presents the CD and CLR of each expressed traffic class as functions of the buffer size under the number of I_3-streams of six. This Figure also shows that the CD increases and the CLR decreases with the buffer size. Moreover, the CD and CLR for I_4-streams are higher than those for I_3-streams. This phenomenon agrees with the results exhibited in Fig. 3 in which more high burstiness traffic incurs lower performance.

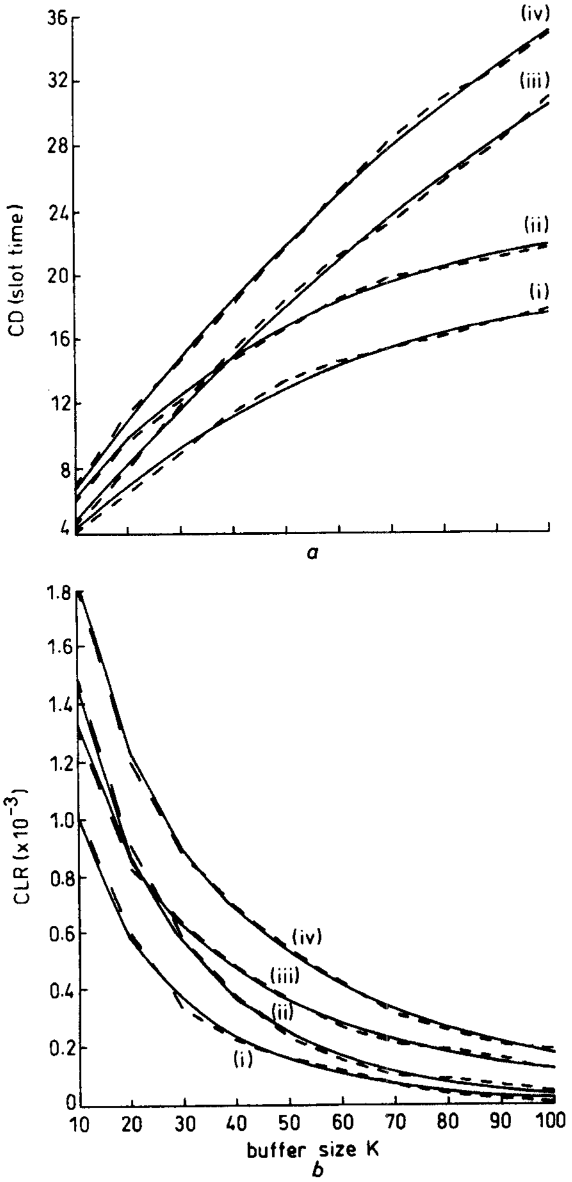


Fig. 5 *CD and CLR as functions of buffer size*
a CD
b CLR
(i) I_3 (ρ = 0.8); (ii) I_4 (ρ = 0.8); (iii) I_3 (ρ = 0.9); (iv) I_4 (ρ = 0.9)
No. of I_3 streams = 6
– – – – simulation
——— analysis

4 QLDC-based CAC algorithm

4.1 QLDC estimation method

Generally, the QLDC method computes the estimated CD (D_i^{QLDC}) and CLR (L_i^{QLDC}) for class-i under an aggregate load of ρ via simple vector multiplication. To illustrate the method efficiently, we examine a system with three traffic classes, resulting in an aggregate load of ρ. Fig. 6 shows the previously derived analytical results of CD for class 1 coexisting with classes 2 and 3. In the Figure, for example, c_{21} represents the CD curve for class 1 under the traffic with dual classes (classes 2 and 1; this is why 'dual-class' in QLDC is so named) and an aggregate load of ρ. Let R_i denote the mean cell arrival rate (cell/slot time) for a class i stream and N_i the number of calls in class i. Thus,

$$N_1R_1 + N_2R_2 + N_3R_3 = \rho \tag{17}$$

In addition, if $N_1 = x$, we obtain

$$c_{21}(x) = r, \quad \text{if } N_1 = x,\ N_2 = (\rho - xR_1)/R_2,\ N_3 = 0 \tag{18}$$

$$c_{31}(x) = s, \quad \text{if } N_1 = x,\ N_2 = 0,\ N_3 = (\rho - xR_1)/R_3 \tag{19}$$

Let l_{j1} (j = 2, 3) represent the line connecting the two ends of c_{j1}. Then, $p(q)$ becomes the value of $l_{21}(l_{31})$ defined at $N_1 = x$, and $u(v)$ becomes the exceeding amount of c_{21} (c_{31}) from l_{21} (l_{31}) at $N_1 = x$. That is, $r = p + u$ and $s = q + v$. Moreover, let Δ_1 denote the maximum exceeding amount of c_{j1} from l_{j1}, for all j, i.e. $\Delta_1 \geq u$ and $\Delta_1 \geq v$. Note that there exists only class 1 in any c_{j1} at $N_1 = \rho/R_1$. Hence all curves c_{j1}, for all j, must intersect at $N_1 = \rho/R_1$ with CD b_1.

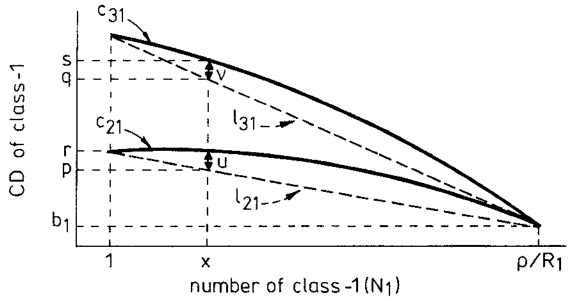


Fig. 6 *CD of class 1 coexisting with class 2 and class 3*

We now assume, under $N_1 = x$ and an aggregate load of ρ, that the CD for class 1 with triple-class traffic is higher than r and less than s. The rationale behind this assumption is that, under $N_1 = x$, class 3 contributes to higher CD for class 1 than class 2 does. As a result, D_1^{QLDC} can be computed on the basis of the proportional CD contribution from dual class 2&1 and dual class 3&1. In addition, we assume that the proportional CD contribution from dual class 2&1 (dual class 3&1) is a linear function of N_2 (N_3) (this is why 'quasilinear' in QLDC is so named). Hence, D_1^{QLDC} defined at $N_1 = x$ becomes

$$D_1^{QLDC} = N_2\frac{p - b_1}{(\rho - xR_1)/R_2} + N_3\frac{q - b_1}{(\rho - xR_1)/R_3} + \Delta_1 + b_1 \tag{20}$$

where the first (second) term represents the proportional CD contribution from class 2 (class 3) to class 1. The third term is added in compensation for the nonlinearity of the curve c_{21} (c_{31}), and the fourth term is the base of all curves c_{j1}, for all j.

Moreover, let a_{j1} denote the gradient of l_{j1}, namely

$$(p - b_1) = (-a_{21})(\rho/R_1 - x) \quad (q - b_1) = (-a_{31})(\rho/R_1 - x) \tag{21}$$

Defining $f_{j1} = (-a_{j1})R_j/R_1$ and by eqns. 20 and 21 one obtains

$$\begin{aligned} D_1^{QLDC} &= (-a_{21})(R_2/R_1)N_2 \\ &\quad + (-a_{31})(R_3/R_1)N_3 + \Delta_1 + b_1 \\ &= f_{21}N_2 + f_{31}N_3 + \Delta_1 + b_1 \\ &= [N_1 \quad N_2 \quad N_3]\begin{bmatrix} 0 \\ f_{21} \\ f_{31} \end{bmatrix} + \Delta_1 + b_1 \end{aligned} \tag{22}$$

So far we have estimated the CD for class 1 under an aggregate load of ρ. Considering the CD for all classes ($1 - C$) under an aggregate load of ρ, $\mathbf{D}^\rho = [D_1^{QLDC}\ D_2^{QLDC} \ldots D_C^{QLDC}]$, we can obtain

$$\mathbf{D}^\rho = \mathbf{N}^\rho \mathbf{F}_D^\rho + \mathbf{T}_D^\rho + \mathbf{B}_D^\rho \qquad (23)$$

where

$$\mathbf{N}^\rho = [N_1 \quad N_2 \quad \cdots \quad N_C] \quad \rho = \sum_{i=1}^{C}(N_i \cdot R_i) \qquad (24)$$

$$\mathbf{F}_D^\rho = [f_{ij}] \qquad f_{ij} = (-a_{ij})R_i/R_j \qquad (25)$$

$$\mathbf{T}_D^\rho = [\Delta_1 \quad \Delta_2 \quad \cdots \quad \Delta_C] \qquad (26)$$

$$\mathbf{B}_D^\rho = [b_1 \quad b_2 \quad \cdots \quad b_C] \qquad (27)$$

Similarly, the CLR for all classes ($1 - C$) under an aggregate load of ρ, $\mathbf{L}^\rho = [L_1^{QLDC}\ L_2^{QLDC} \ldots L_C^{QLDC}]$, can be estimated by

$$\mathbf{L}^\rho = \mathbf{N}^\rho \mathbf{F}_L^\rho + \mathbf{T}_L^\rho + \mathbf{B}_L^\rho \qquad (28)$$

where $\mathbf{F}_L^\rho$, $\mathbf{T}_L^\rho$ and $\mathbf{B}_L^\rho$ are to CLR as $\mathbf{F}_D^\rho$, $\mathbf{T}_D^\rho$ and $\mathbf{B}_D^\rho$ are to CD. Finally, note that $\mathbf{F}_D^\rho$, $\mathbf{T}_D^\rho$, $\mathbf{B}_D^\rho$, $\mathbf{F}_L^\rho$, $\mathbf{T}_L^\rho$ and $\mathbf{B}_L^\rho$, for all ρ, can be computed in advance and stored for later performing of the CAC algorithm.

4.2 QLDC-based CAC algorithm

Based on the QLDC estimation method, we now present the proposed CAC algorithm.

Algorithm:

1. For each class i, compute the resulting number of calls in $(t-1, t]$, N_i, where

   N_i = (the number of existing calls for class i prior to time $t-1$)

   + (the number of newly requesting calls for class i within $(t-1, t]$)

   – (the number of terminating calls for class i within $(t-1, t]$).

2. From the storage, fetch $\mathbf{F}_D^\rho$, $\mathbf{T}_D^\rho$, $\mathbf{B}_D^\rho$ $\mathbf{F}_L^\rho$, $\mathbf{T}_L^\rho$ and $\mathbf{B}_L^\rho$, where $\rho = \Sigma_{i=1}^{C}(N_i \cdot R_i)$.

3. Compute D_i^{QLDC} and L_i^{QLDC} for all i, from eqns. 23 and 28.

4. Reject all newly requesting calls if delay or loss QoS of any existing calls for class i cannot be satisfied; then go to step 6.

5. Pass each newly requesting call to the next node along the path toward the destination, and make call acceptance decisions by the same procedure shown from step 1 to step 4.

6. Return to step 1 at the beginning of the $(t+1)$th slot time.

In step 5 of the algorithm, the CD and CLR estimated at intermediate nodes are based on the mean burst length observed at the source node. It is worth noting that the greater the mean burst length the higher the CD and CLR. In addition, the mean burst length of a call at the $(n+1)$th node is smaller than that at the nth node [18] due to the mixing and interleaving of calls along the path. Consequently, our CAC algorithm makes call acceptance decisions in a *conservative* manner.

4.3 QLDC-based results

To demonstrate the accuracy of the QLDC-based CAC algorithm, we draw comparisons between QLDC-based results and simulation results for I_3-streams under five single node cases (see Table 2), and five end-to-end cases (see Table 3) based on the network model shown in Fig. 7. In each single node case the traffic is composed of four traffic classes (I_1, M_2, I_3 and I_4), resulting in an aggregate load of 0.9. Moreover, the buffer size K is assumed to be 100. In Tables 2 and 3, D_3^{sim} and L_3^{sim} represent the simulation results of CD and CLR, respectively.

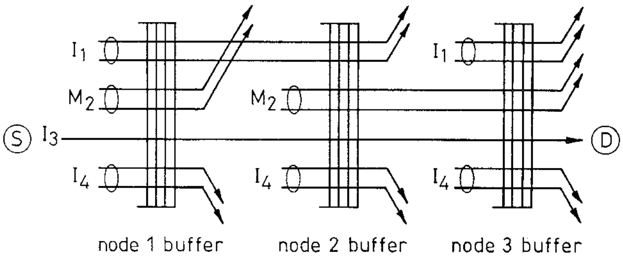


Fig.7 *Network model of end-to-end cases*

We now illustrate the computation of D_i^{QLDC} and L_i^{QLDC} in end-to-end case 5. To compute D_i^{QLDC} and L_i^{QLDC} we first obtain $[\mathbf{F}_{D(3)}^{0.9}\ \mathbf{F}_{L(3)}^{0.9}]$, $[\mathbf{F}_{D(3)}^{0.6}\ \mathbf{F}_{L(3)}^{0.6}]$, $[\mathbf{F}_{D(3)}^{0.3}\ \mathbf{F}_{L(3)}^{0.3}]$, $[\mathbf{T}_{D(3)}^{0.9}\ \mathbf{T}_{L(3)}^{0.9}]$, $[\mathbf{T}_{D(3)}^{0.6}\ \mathbf{T}_{L(3)}^{0.6}]$, $[\mathbf{T}_{D(3)}^{0.3}\ \mathbf{T}_{L(3)}^{0.3}]$,

Table 2: Summary of single node cases

	N_1	N_2	N_3	N_4	D_3^{QLDC}	D_3^{sim}	L_3^{QLDC}	L_3^{sim}
Case 1	1	10	2	6	3.187 × 10	3.084 × 10	1.318×10^{-2}	1.250×10^{-2}
Case 2	3	5	3	4	3.067 × 10	3.024 × 10	9.864×10^{-3}	9.697×10^{-3}
Case 3	4	15	1	3	2.864 × 10	2.836 × 10	8.162×10^{-3}	7.908×10^{-3}
Case 4	5	5	3	2	2.890 × 10	2.790 × 10	6.530×10^{-3}	6.524×10^{-3}
Case 5	3	5	5	3	2.870 × 10	2.859 × 10	7.780×10^{-3}	7.559×10^{-3}

Table 3: Summary of end-to-end cases

	Node 1					Node 2					Node 3					D_3^{QLDC}	D_3^{sim}	L_3^{QLDC}	L_3^{sim}
	N_1	N_2	N_3	N_4	ρ	N_1	N_2	N_3	N_4	ρ	N_1	N_2	N_3	N_4	ρ	(10^1)	(10^1)	(10^{-2})	(10^{-2})
Case 1	1	45	1	3	0.9	1	5	1	7	0.9	2	5	1	6	0.9	8.989	8.885	3.530	3.366
Case 2	3	15	1	4	0.9	3	5	1	2	0.6	4	5	1	4	0.9	6.709	6.418	2.007	1.969
Case 3	1	15	1	6	0.9	1	25	1	2	0.6	2	25	1	1	0.6	4.067	3.997	1.325	1.317
Case 4	2	25	1	1	0.6	2	5	1	6	0.9	1	5	1	1	0.3	3.980	3.772	1.364	1.306
Case 5	1	5	1	1	0.3	1	15	1	3	0.6	2	15	1	5	0.9	3.822	3.820	1.159	1.146

$[\mathbf{B}^{0.9}_{D(3)}\ \ \mathbf{B}^{0.9}_{L(3)}]$, $[\mathbf{B}^{0.6}_{D(3)}\ \ \mathbf{B}^{0.6}_{L(3)}]$ and $[\mathbf{B}^{0.3}_{D(3)}\ \ \mathbf{B}^{0.3}_{L(3)}]$ under aggregate loads of 0.9, 0.6 and 0.3, respectively (as shown in Section 4.3.1). Thus, according to eqns. 23 and 28, D_i^{QLDC} and L_i^{QLDC} can be computed as:

$$\begin{aligned}
D_3^{QLDC} &= [1 \quad 5 \quad 1 \quad 1]\,\mathbf{F}^{0.3}_{D(3)} + \mathbf{T}^{0.3}_{D(3)} + \mathbf{B}^{0.3}_{D(3)} + \\
&\quad [1 \quad 15 \quad 1 \quad 3]\,\mathbf{F}^{0.6}_{D(3)} + \mathbf{T}^{0.6}_{D(3)} + \mathbf{B}^{0.6}_{D(3)} + \\
&\quad [2 \quad 15 \quad 1 \quad 5]\,\mathbf{F}^{0.9}_{D(3)} + \mathbf{T}^{0.9}_{D(3)} + \mathbf{B}^{0.9}_{D(3)} \\
&= 3.822e + 1 \\
L_3^{QLDC} &= 1 - \Big\{ \Big(1 - [1 \quad 5 \quad 1 \quad 1]\,\mathbf{F}^{0.3}_{L(3)} - \mathbf{T}^{0.3}_{L(3)} - \mathbf{B}^{0.3}_{L(3)}\Big) \\
&\quad \cdot \Big(1 - [1 \quad 15 \quad 1 \quad 3]\,\mathbf{F}^{0.6}_{L(3)} - \mathbf{T}^{0.6}_{L(3)} - \mathbf{B}^{0.6}_{L(3)}\Big) \\
&\quad \cdot \Big(1 - [2 \quad 15 \quad 1 \quad 5]\,\mathbf{F}^{0.9}_{L(3)} - \mathbf{T}^{0.9}_{L(3)} - \mathbf{B}^{0.9}_{L(3)}\Big) \Big\} \\
&= 1.159e - 2
\end{aligned}$$

Furthermore, Figs. 8 and 9 show the performance discrepancy by means of bar charts. In both Figures we have discovered that our QLDC-based results agree with simulation results with a discrepancy of as low as 0.06 for both the single node and end-to-end cases.

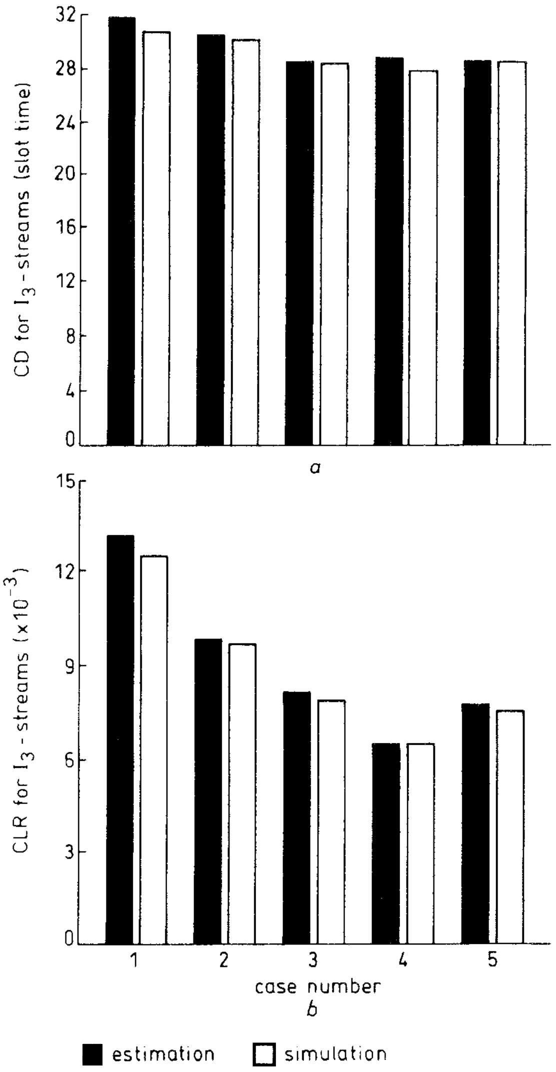


Fig. 8 *Performance discrepancy under single node cases*
Traffic is composed of four classes: I_1, M_2, I_3, I_4
a CD
b CLR

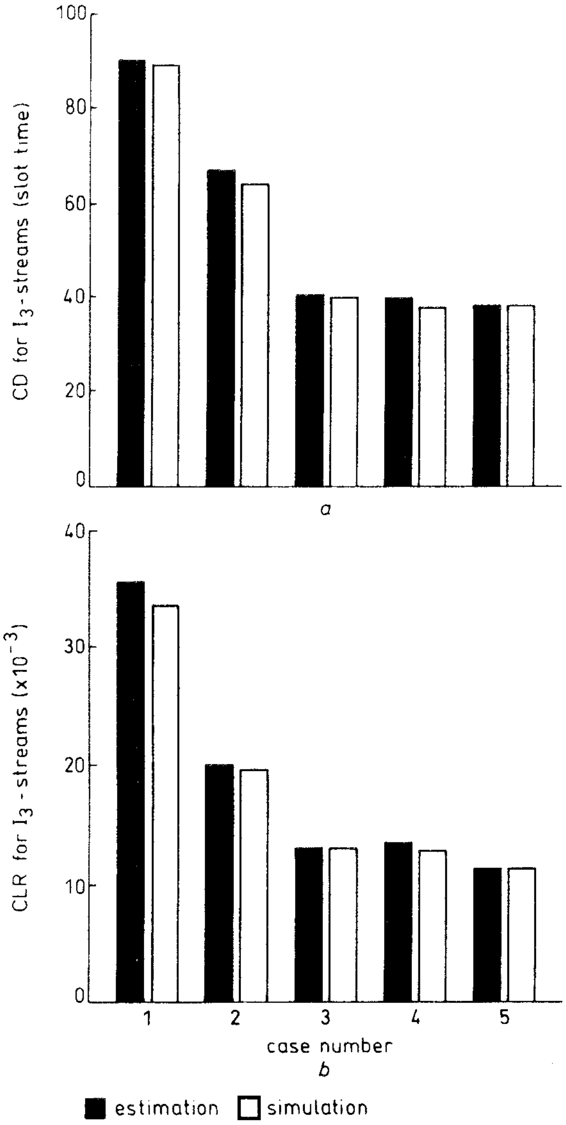


Fig. 9 *Performance discrepancy under end-to-end cases*
Traffic is composed of four classes: I_1, M_2, I_3, I_4
a CD
b CLR

4.3.1 Summary of matrices and vectors of end-to-end cases:

$$[\mathbf{F}^{0.9}_{D(3)}\mathbf{F}^{0.9}_{L(3)}] = \begin{bmatrix} 1.07691 & 4.16646 \times 10^{-4} \\ -1.14496 \times 10^{-1} & -3.49956 \times 10^{-6} \\ 0 & 0 \\ 1.96264 & 2.08340 \times 10^{-3} \end{bmatrix}$$

$$[\mathbf{F}^{0.6}_{D(3)}\mathbf{F}^{0.6}_{L(3)}] = \begin{bmatrix} 2.42517 \times 10^{-1} & 5.40654 \times 10^{-8} \\ -1.89250 \times 10^{-2} & -1.55322 \times 10^{-12} \\ 0 & 0 \\ 7.93851 \times 10^{-1} & 3.09780 \times 10^{-5} \end{bmatrix}$$

$$[\mathbf{F}^{0.3}_{D(3)}\mathbf{F}^{0.3}_{L(3)}] = \begin{bmatrix} 4.73794 \times 10^{-2} & 2.51275 \times 10^{-13} \\ -4.91348 \times 10^{-3} & -1.76122 \times 10^{-18} \\ 0 & 0 \\ 1.76152 \times 10^{-1} & 2.79002 \times 10^{-9} \end{bmatrix}$$

$$[\mathbf{T}^{0.9}_{D(3)}\mathbf{T}^{0.9}_{L(3)}] = [\ 3.40680 \quad 1.71491 \times 10^{-20}]$$

$$[\mathbf{T}^{0.6}_{D(3)}\mathbf{T}^{0.6}_{L(3)}] = [\ 2.81174 \times 10^{-2} \quad 0]$$

$$[\mathbf{T}^{0.3}_{D(3)}\mathbf{T}^{0.3}_{L(3)}] = [\ 2.38818 \times 10^{-5} \quad 1.34326 \times 10^{-33}]$$

$$[\mathbf{B}^{0.9}_{D(3)}\mathbf{B}^{0.9}_{L(3)}] = [\ 1.67508 \times 10 \quad 2.97481 \times 10^{-4}]$$

$$[\mathbf{B}^{0.6}_{D(3)}\mathbf{B}^{0.6}_{L(3)}] = [\ 3.57929 \quad 8.54273 \times 10^{-11}]$$

$$[\mathbf{B}^{0.3}_{D(3)}\mathbf{B}^{0.3}_{L(3)}] = [\ 1.66974 \quad 4.40304 \times 10^{-17}]$$

4.4 Time and space complexity

The time complexity of the QLDC-based CAC algorithm at a node is clearly dominated by eqns. 23 and 28, requiring $2C$ vector multiplications. The time complexity of the algorithm is thus $O(C)$ vector multiplications. Besides, the space complexity of the algorithm at a single node is dominated by $\mathbf{F}_D{}^{\rho}$ and $\mathbf{F}_L{}^{\rho}$, for all ρ. Since the dimension of $\mathbf{F}_D{}^{\rho}$ for any given ρ is C^2 subject to the total bandwidth being divided into W aggregate load levels, the space complexity of the algorithm is thus $O(WC^2)$ bytes.

To illustrate the viability of the CAC algorithm, let us examine the example given as follows. Assume that the capacity of each physical link is 1Gbit/s (i.e. 2^{30}bit/s), the capacity of each channel is 64Kbit/s (i.e. 2^{16}bit/s), the total number of traffic classes is 64, and each element of $\mathbf{F}_D{}^{\rho}$ takes up a storage of 2 bytes. Thus, the total number of aggregate load levels, W, becomes $2^{30}/2^{16} = 2^{14}$. Consequently, the total space required by $\mathbf{F}_D{}^{\rho}$, $\mathbf{T}_D{}^{\rho}$, $\mathbf{B}_D{}^{\rho}$, $\mathbf{F}_L{}^{\rho}$, $\mathbf{T}_L{}^{\rho}$ and $\mathbf{B}_L{}^{\rho}$, for all ρ, is $2^{14}(64^2+64+64)\cdot 2\cdot 2 \approx 257$M bytes, which is rational with respect to the cost and table look-up overhead in ATM switches.

5 Conclusions

The goal of the paper has been the provision of an efficient CAC algorithm based on a so-called quasilinear dual-class correlation (QLDC) estimation method. We initially provided an analysis of the cell delay and cell loss ratio for each traffic class based on a queuing system with dual arrivals (Bernoulli and IBP). The paper showed the accuracy of the analysis via simulations. We also observed that more high burstiness traffic in a switch incurred a decrease of the statistical multiplexing gain. According to the analysis, we proposed QLDC, which conservatively estimated both the cell delay and cell loss ratio for each traffic class via simple vector multiplication. We then presented our QLDC-based CAC algorithm. Numerical results exhibited that our QLDC-based results agreed with simulation results with a discrepancy of as low as 0.06 for both the single node and end-to-end cases. Finally, we justified the viability of the CAC algorithm in ATM switches by showing that the algorithm incurred low time complexity $O(C)$ (in vector multiplications) and space complexity $O(WC^2)$ (in bytes), where C is the total number of traffic classes and W is the total number of aggregate load levels.

6 References

1 HÄNDEL, R., HUBER, M.N., and SCHRÖDER, S.: 'ATM networks—concept, protocol, applications' (Addison-Wesley, 1993, 2nd edn.)

2 PRYCKER, M.D.: 'Asynchronous transfer mode solution for broadband ISDN' (Ellis Horwood, 1993)

3 HÄNDEL, R., and HUBER, M.N.: 'Integrated broadband networks: an introduction to ATM-based networks' (Addison-Wesley, 1991)

4 SAITO, H.: 'Teletraffic technologies in ATM networks' (Artech House, 1994)

5 YATES, D., KUROSE, J., TOWSLEY, D., and HIUCHYJ, M.G.: '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 CHEN, X.: 'Modeling connection admission control'. IEEE INFOCOM'93, 1993, pp. 274–281

7 CHO, K.-T., and KAWASAKI, S.-K.: 'Call admission control method in ATM networks'. IEEE ICC'92, 1992, pp. 1628–1633

8 SOHRABY, K.: '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 ELWALID, A.I., and MITRA, D.: 'Effective bandwidth of general Markovian traffic sources and admission control of high speed networks'. IEEE INFOCOM'93, 1993, pp. 256–265

10 SAITO, H., and SHIOMOTO, K.: 'Dynamic call admission control in ATM networks', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (7), pp. 982–989

11 LEE, T.H., LAI, K.C., and DUANN, S.-T.: 'Real time call admission control for ATM networks with heterogeneous bursty traffic'. IEEE ICC'94, 1994, pp. 80–85

12 GIBBENS, R.J., KELLY, F.P., and KEY, P.B.: 'A decision-theoretic approach to call admission control in ATM networks', *IEEE J. Sel. Areas Commun.*, 1995, **13**, (6), pp. 1101–1113

13 MURASE, T., SUZUKI, H., SATO, S., and TAKEUCHI, T.: 'A call admission control algorithm for ATM network using a simple quality estimate', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (9), pp. 1461–1470

14 DAILIANAS, A., and BOVOPOULOS, A.: 'Real-time admission control algorithm with delay and loss guarantee in ATM networks'. IEEE INFOCOM'94, 1994, pp. 1065–1072

15 LIN, F.Y.S., and YEE, J.R.: 'A real-time distributed routing and admission control algorithm for ATM networks'. IEEE INFOCOM'93, 1993, pp. 792–801

16 SUZUKI, H., MURASE, T., SATO, S., and TAKEUCHI, T.: 'A burst traffic control strategy for ATM networks'. Proc. IEEE GLOBECOM'90, 1990, pp. 505.6.1–505.6.5

17 RATHGEB, E.P.: 'Modeling and performance comparison of policing mechanisms for ATM networks', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (3), pp. 325–334

18 OHBA, Y., MURATA, M., and MIYAHARA, H.: 'Analysis of interdeparture processes for bursty traffic in ATM networks', *IEEE J. Sel. Areas Commun.*, 1991, **9**, (3), pp. 468–476

19 HUI, J.Y.: 'Resource allocation for broadband networks', *IEEE J. Sel. Areas Commun.*, 1988, **6**, (9), pp. 1598–1608

20 DAIGLE, J.N.: 'Queueing theory for telecommunications' (Addison–Wesley, 1992)

21 'MATLAB: high-performance numeric computation and visualization software'. The MATH WORKS Inc., August 1992