

Steady-state analysis of rate-controlled data transport in ATM Local Area Networks

Hong-Bin Chiou^{1,a}, Zsehong Tsai^b

^aTelecommunication Laboratories, Chunghwa Telecomm. Co. Ltd, Taipei, Taiwan

^bDepartment of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Received 2 February 1996; accepted 29 May 1996

Abstract

This paper presents a steady-state analysis of rate-based flow control operating with the regular window mechanism of the transport protocol in ATM local area network (LAN) or local high-speed network environments. The rate control schemes are divided into single-threshold and two-threshold cases. By applying Norton's theorem for queuing model simplification and deriving a computation algorithm, the analytical results for steady-state performance of these schemes are found to be computationally efficient and yield accurate approximations. In the numerical results, the rate control schemes have been shown to have significant effects in reducing congestion levels at the receiving entity as well as in the high-speed networks. We also conclude that the two-threshold rate control scheme can be more effective in alleviating network congestion conditions than single-threshold control, while keeping the control overhead (in terms of the rate adjustment frequency) at a low level, provided that threshold values are selected appropriately. The total throughput could be affected, but only in a limited fashion. © 1997 Elsevier Science B.V.

Keywords: ATM; Rate Control; LAN

1. Introduction

In recent years the development of high-speed networks has triggered the re-examination of the applicability of existing control techniques in this new environment, and has led to a series of studies in flow and congestion control issues. The two major areas in these studies are window-based and rate-based flow control. For the widely studied static window-based models, readers are referred to previous works [1–3]. However, it has been pointed out in Ref. [2] that performance of window-based transport protocols can be easily degraded in a high-speed environment. In turn, rate-based flow control schemes [4–7], especially feedback-based approaches, have attracted much more attention and seem more promising for high-speed networks, especially for ATM LANs.

Rate control schemes that operate on an end-to-end basis are easier to implement in the higher protocol layers, especially when end-to-end transport is considered [8–10]. There are two typical approaches in the implementation of

a rate control scheme. The first approach is to control the amount of data transported across the network, which is also called “credit” [9], while the second approach is to maintain a minimal departure time interval [8]. However, rate control schemes alone cannot provide all the control functions required in a transport protocol. Since the acknowledgment operations in window flow control can be used together with error-control schemes such as Go-back-N and selective repeat ARQ, both the rate and the window mechanism are included in protocols like XTP [9] and NETBLT [10]. Although combined flow control techniques are more complicated, we believe such control methods in the transport protocol will become more popular in the future, so that related performance issues deserve detailed investigation.

In the following, we first give a brief review of existing analytical results of multilayered models for window flow control and theoretical models for rate control schemes to illustrate the need for such a framework.

With respect to the analysis of multilayered communication protocols, a typical approach is to construct a queuing network model based directly on the structure of the ISO reference model, and then to solve the resulting model using an iterative approximation procedure and existing formulas for BCMP² networks [11]. For details of networks in various

¹ Present address: Electrical Engineering Department, National Taiwan University, Taipei, Taiwan. Fax: + 886 2 755 1982; e-mail: chiu@eagle.ntu.edu.tw

² Queuing networks with product-form solutions were first proposed by Baskett et al. [22].

LAN environments, see Refs. [12,13]. More recent results can be found in Fdida et al. [14], who present a methodology for analyzing networks controlled by a nested sliding window mechanism, whereby each layer of sliding window control can be simplified to a state-dependent infinite server queue using a flow-equivalent method. Also, Shapiro and Perros [15] present a hierarchical method for analyzing nested sliding windows in a multilayered environment, paying specific attention to the analysis of packet fragmentation and reassembly behavior. To facilitate the analysis of multi-layer operations, an equivalent queue methodology is also applied. However, in Ref. [15] no effort is made to extend the models to cover the case of rate control schemes. The combination of both window and rate control schemes is discussed only in Refs. [9,17] with no analytical results.

With rate control schemes, analytical results for models which employ queue size as a congestion indicator can be found in Refs. [6,7,18,19]. A simple analytical model with a single source is solved in Ref. [6] to observe the time-dependent behavior of the sending rate and the queue size at the bottleneck in both steady state and transient regimes. A feedback control policy in a packet switching network with significant propagation delay is investigated in Ref. [7], where the behavior of each individual source can be analyzed in a multiple virtual circuit environment. In Ref. [18] the network background traffic is modeled as the disturbance in the service rate of the congested node using a fluid flow model. In Ref. [19] a two-threshold feedback rate control policy is investigated. Among these papers, only Ref. [19] employed a discrete-time Markov chain, and the differential equation technique remains the most popular analytical approach. In Refs. [6,7,18,19], analytical results were successfully used to illustrate various behaviors of rate-based flow control. However, none of these results can be easily extended to queueing models which incorporate a window mechanism such as those in the high-speed transport protocols; and none of the models are used to characterize the behavior of those rate control schemes via message interval control. Hence, we can conclude that a framework for the queueing analysis of a transport protocol that can incorporate both window-based and rate-based flow control still merits further investigation.

In this paper we study the steady-state performance of typical rate control schemes that can be employed in a window mechanism of the transport protocol in ATM LANs. These rate control schemes are based on message transmission interval enforcement and operate with single-threshold or two-threshold options for congestion feedback mechanisms at the message level. In more recent developments of ATM switches, the per-VC queueing and packet-based buffer management policies, such as early packet discarding [16], have also become popular. All of these new trends indicate that such a message-based control scheme could be promising in future high-speed LANs.

In Section 2 we describe in detail a queueing model and the operation of a generic rate-controlled window scheme

that fit the needs of data transport in ATM LANs. In Section 3 we present a queueing analysis for single-threshold and two-threshold rate control. In Section 4, numerical results and validation, we provide approximate results as well as simulation statistics. Finally, in Section 5 we present our conclusions.

2. The rate control scheme in a transport layer

In this section we describe briefly how a typical rate control scheme behaves when it is incorporated into a window mechanism in the transport protocol. Details of the use of such an integration of both window and rate control schemes are given in Refs. [9,10]. We consider a network environment in which two transport layer entities on separate end-systems are connected to an ATM LAN switch. The end-systems assume a multiple-layer protocol stack over ATM, while the transport protocol operating on these two transport entities performs end-to-end flow control. The end-to-end flow control protocol is assumed to be transparent to the lower layer protocol. Briefly, there are two major flow control mechanisms operating simultaneously in this transport protocol: window-based and rate-based control. This window mechanism imposes no constraints on the rate at which messages can be sent.

The rate-based control scheme ensures that the interval between two consecutive message transmissions is larger than or equal to a minimum gap. The operation of the rate-controlled window scheme is as follows. When the sending transport layer entity, called simply the sending entity, has messages to transmit, a timer is used to control the interval between the starting instants of two consecutive message transmissions, so that messages can not be sent by the sending entity until the timer expires. In contrast with traditional window control, the message must be held for a period of time even if the current residual window size is not equal to zero. As in legacy window schemes, every message is acknowledged by the receiving entity via the other link in the reverse direction.

In addition, the ATM LAN switches are responsible for reporting network congestion to the sending entity by monitoring the number of messages queued in switches. When the number of messages in the switch exceeds a congestion indication threshold, a congestion condition is signalled to the lower-layer entity of the end system by a special RM (resource management) cell. The congestion indication is then passed to the transport layer entity. Alternatively, when the feedback information reflects only the cell-level congestion information, the lower-layer entity is responsible for interpretation and should translate it to the message-level congestion indicator. After the lower-entity reports congestion indication to its upper layer, the message sending rate is lowered. Only when the message queue is reduced below a congestion relief threshold, can the message sending rate return to its original rate. In our scheme,

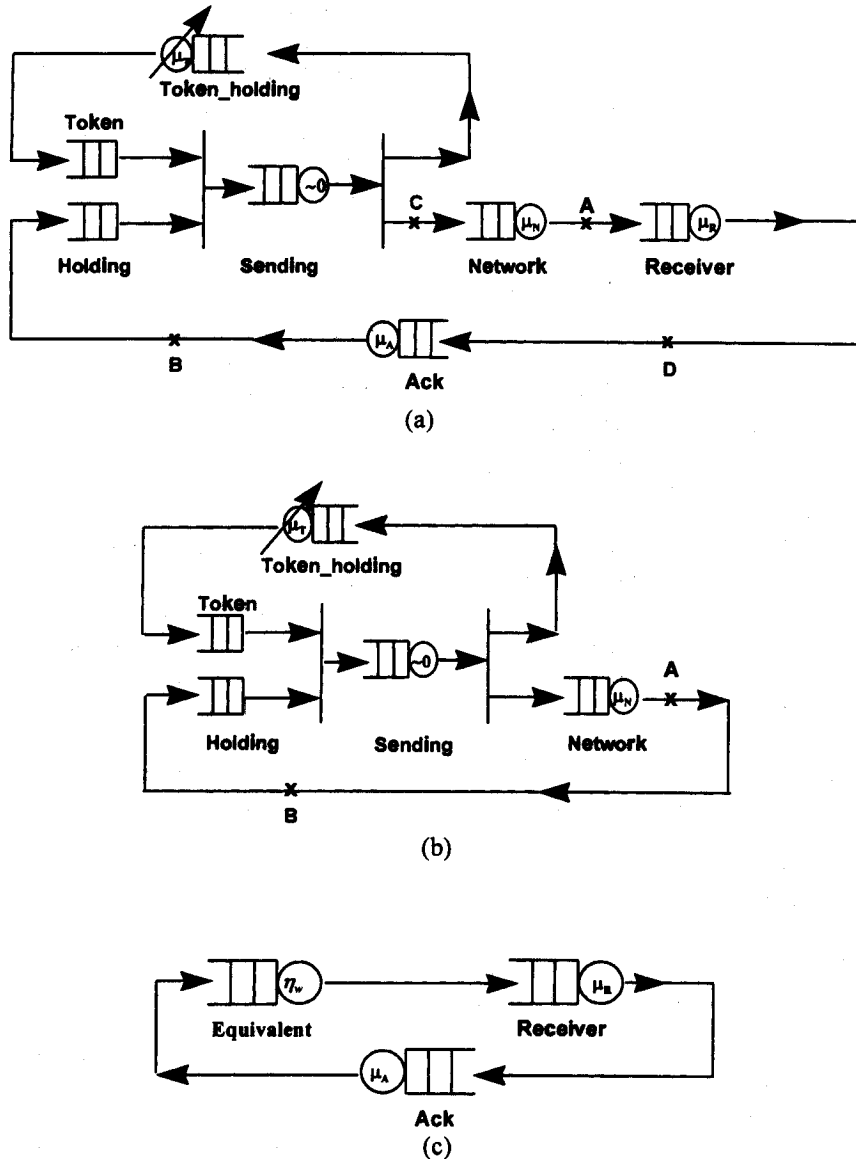


Fig. 1. (a) Original queuing model, (b) shorted queuing network, and (c) equivalent queuing network.

we call it a single-threshold case if the congestion indication threshold is the same as the relief threshold; otherwise, it is called the two-threshold case.

3. The queuing model

Since our objective is to model the behavior of the transport protocol during heavy traffic conditions, we employ a closed queuing network model. Similarly, we assume that the sender is always ready to transmit. Considering that a transport protocol must perform error control functions using the same window mechanism, we assume that the underlying lower-layer protocol and ATM LAN provides sufficiently reliable transmission so that the window mechanism performs only flow control functions instead of error control. The loss of messages thus has a negligible

impact on transport protocol performance. We assume that lower-layer protocol processors have infinite buffers and their probability of message loss is zero. In the following, we describe such a closed queuing network model for our rate-controlled window mechanism.

The queuing network is shown in Fig. 1(a). In order to represent such a queuing model in a simple manner, we use two symbols, *Join* and *Split*, as shown in Fig. 2. Seven queues in total are included in this model: *Token*, *Token_holding*, *Holding*, *Sending*, *Network*, *Receiver* and *Ack*. Obviously, here the *Network* and the *Ack* queues represent the equivalent queues in the forward and reverse directions of the underlying network service (including network layer, AAL and ATM layers, etc.), respectively. The protocol processor of the receiving entity is represented by the *Receiver* queue, while other queues are used to model the operations of the sending entity. Two classes of customer

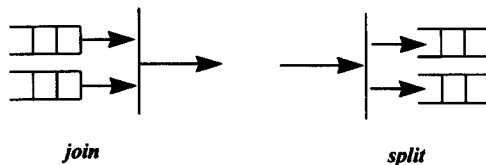


Fig. 2. Symbols for *Join* and *Split*.

are assumed: token and messages. Two separate loops coexist in this model, the token loop and the message loop. The token loop is the loop where a single token circulates, representing the behavior of the rate control mechanism. The token starts from *Token_holding*, moves to *Token*, jointly enters the *Sending* queue with a message, and then goes back to *Token_holding*. Within this loop, the *Token_holding* queue characterizes the rate control timer, and the *Join* node characterizes the rate control operation. In the message loop, messages circulate in the follow fashion: starting at the *Holding* queue, they move to *Sending* if a token is available, visit *Network*, *Receiver*, *Ack*, and then return to the *Holding* queue. The number of messages circulating in the message loop is equal to window size W . The detailed operations are as follows.

When a message arrives at the *Holding* queue and finds a token available in the *Token* queue, the message leaves the *Holding* queue and takes the token out from the *Token* queue at the same time. This is called the *Join* operation. The message is then allowed to enter the *Sending* queue. If the *Token* queue is empty when a message arrives at the *Holding* queue, the message needs to be held until the token enters the *Token* queue. The service time of the message in the *Sending* queue represents the time required to complete the transmission of that message by the sending entity. When the message leaves the *Sending* queue, it enters the *Network* queue and places a token into the *Token_holding* queue at the same time. This is called the *Split* operation. After an interval of length T , the token departs from the *Token_holding* queue and enters the *Token* queue again. The mean token sojourn time in the *Token_holding* queue is allowed to alternate dynamically between high and low values, depending on the feedback information. The feedback information sent from the *Network* queue, the congestion indication, is assumed to be received by the *Token_holding* queue immediately. Hence, the *Token_holding* queue can monitor the *Network* queue constantly via this feedback information. When the number of messages in the *Network* queue exceeds the threshold, it increases the mean token sojourn time to reduce the message sending rate. In turn, when the number of messages in the *Network* queue is below the threshold, the message sending rate is set at the high value again. We assume that the update of the sending rate at the *Token_holding* queue is operated in a non-preemptive mode. In other words, the sojourn time of a currently held message in the *Token_holding* queue remains unchanged when the feedback information arrives. Two possible mean token sojourn times,

denoted $1/\mu_H$ and $1/\mu_L$, are employed for the high and low sending rates, respectively. When $\mu_L = \mu_H$, the rate control scheme reduces to the single-rate case.

Although the propagation delay is assumed to be zero in this model, our non-preemptive operation can actually absorb the effect of non-zero feedback delay, if the feedback delay is relatively small compared with the message transmission time. Hence our model should be able to illustrate rate control behavior in a local environment.

In our analytical model the token sojourn time at the *Token_holding* queue, or equivalently, the rate control time T , is assumed to be a random variable with Erlang- k distribution, which should yield a good approximation to the constant timer behavior, especially when the value of k is large. As for the *Sending* queue, we model it as an FCFS single server with service time equal to zero. *Network*, *Receiver* and *Ack* queues, which model other network components, are assumed to have FCFS single server and exponential service time. Note that the exponential service time assumption of the *Network* queue actually allows easy application of the equivalent queue method for multilayered modeling [15]. We use the regular assumption that all service time distributions of all queues are independent random variables. Last, but not least, all queues are assumed to have infinite buffers so that no buffer overflow can occur.

In this model we employ only a single server queue, the *Network* queue, to model the lower layer operations of both end systems and the high-speed ATM LAN. However, according to Ref. [15] and our discussion in Section 1, our queueing network can be used as a queueing model to illustrate the queueing behavior in various multilayered communication environments when the upper layer protocol employs a rate-controlled window scheme. Similar models, described in Ref. [3], employ different forms of equivalent queues to model the behavior of a network with background traffic. We therefore believe that our closed queueing network could actually represent a large class of high-speed networks.

3.1. Queueing analysis: single-threshold rate control

We introduce here an approach for the efficient calculation of steady-state probability of our rate-controlled window scheme model for the single-threshold case. The definitions of the required notations are as follows:

n_H :	number of messages in the <i>Holding</i> queue.
n_S :	the current phase at which the token stays in the <i>Token_holding</i> queue, $1 \leq n_S \leq k$
n_N :	number of messages in the <i>Network</i> queue.
μ_T :	service rate in the <i>Token_holding</i> queue, under single-rate mode with $\mu_T = \mu_S$.
μ_A :	service rate in the <i>Ack</i> queue.
μ_R :	service rate in the <i>Receiver</i> queue.
μ_N :	service rate in the <i>Network</i> queue.
k :	the number of phases for the Erlang- k distributed sojourn time in the <i>Token_holding</i> queue.
L :	threshold of the queue length in the <i>Network</i> queue.
M :	operation mode of the <i>Token_holding</i> queue, $M = 0$ or 1 .

Two steps are required in the performance analysis. First, we employ the flow equivalent server method to simplify the model shown in Fig. 1(a) in a way similar to the application of Norton's theorem³ [20,21]. The procedure for this simplification is as follows. We short points A and B of the model in Fig. 1(a) to get the shorted queueing network in Fig. 1(b). For this queueing model, system state (i,j,h) stands for the state of the system when $n_H = i$, $n_S = j$ and $M = h$, where $M = 0$ means that the *Token_holding* queue is operated in the low sending rate mode, while $M = 1$ indicates the high sending rate mode. P_{idle} and $P_{i,j,h}$ stand for stationary probabilities of the (*idle*) and (i,j,h) states, respectively. The total number of messages in the shorted queueing network in Fig. 1(b) is equal to w , where $w \leq W$. In addition, two indicator functions, $I_L(i)$ and $\bar{I}_L(i)$, are defined as follows:

$$I_L(i) = \begin{cases} 1, & i = L \\ 0, & \text{otherwise} \end{cases} \quad \bar{I}_L(i) = \begin{cases} 1, & \text{otherwise} \\ 0, & i = L \end{cases}$$

The global balance equations for the shorted queueing network in Fig. 1(b) are as follows. Because system states $(0 \leq i \leq w - L, j, 1)$ do not exist, they are not listed in the global balance equations.

For $n_H = 0$,

$$\begin{cases} (\mu_N + k\mu_L)P_{0,j,0} = k\mu_L P_{0,j+1,0}, & 1 \leq j \leq k-1 \\ (\mu_N + k\mu_L)P_{0,j,0} = k\mu_L P_{1,1,0} + \mu_N P_{idle}, & j = k \\ \mu_N P_{idle} = k\mu_L P_{0,1,0} \end{cases}$$

For $1 \leq n_H < w - L + 1$,

$$\begin{cases} (\mu_N + k\mu_L)P_{i,j,0} = k\mu_L P_{i,j+1,0} + \mu_N P_{i-1,j,0}, & 1 \leq j \leq k-1 \\ (\mu_N + k\mu_L)P_{i,j,0} = k\mu_L P_{i+1,1,0} + \mu_N P_{i-1,0} + I_{w-L+1}(i+1)k\mu_H P_{i+1,1,1}, & j = k \end{cases}$$

For $w - L + 1 \leq n_H < w$,

$$\begin{cases} (\mu_N + k\mu_L)P_{i,j,0} = k\mu_L P_{i,j+1,0} + \mu_N P_{i-1,0}, & 1 \leq j \leq k-1 \\ (\mu_N + k\mu_L)P_{i,j,0} = \mu_N P_{i-1,0}, & j = k \\ (\mu_N + k\mu_H)P_{i,j,1} = k\mu_H P_{i,j+1,1} + \bar{I}_{w-L}(i-1)\mu_N P_{i-1,j,1}, & 1 \leq j \leq k-1 \\ (\mu_N + k\mu_H)P_{i,j,1} = k\mu_H P_{i+1,1,1} + k\mu_L P_{i+1,1,0} + \bar{I}_{w-L}(i-1)\mu_N P_{i-1,j,1}, & j = k \end{cases}$$

For $n_H = w$,

$$\begin{cases} k\mu_L P_{w,j,0} = k\mu_L P_{w,j+1,0} + \mu_N P_{w-1,j,0}, & 1 \leq j \leq k-1 \\ k\mu_L P_{w,j,0} = \mu_N P_{w-1,j,0}, & j = k \\ k\mu_H P_{w,j,1} = k\mu_H P_{w,j+1,1} + \mu_N P_{w-1,j,1}, & 1 \leq j \leq k-1 \\ k\mu_H P_{w,j,1} = \mu_N P_{w-1,j,1}, & j = k \end{cases}$$

The state transition diagram in Fig. 3 is based on the global balance equations. In Fig. 3, system states $(i,j,0)$, which stand for the *Token_holding* queue operating in low sending rate mode, are located on the upper plane, while system

states $(i,j,1)$, which stand for the *Token_holding* queue operating in high sending rate mode, are located on the lower plane. After examining this state transition diagram, one may conclude that the underlying system process is still not time-reversible. Hence, a product-form solution can not be easily obtained since detailed balance equations do not hold. Fortunately, the structure of this diagram follows certain rules. For example, when the stationary probability of $P_{0,1,0}$ is known, its neighboring states *idle* and $P_{0,2,0}$ can be solved. We have derived a computational algorithm which can solve the global balance equations effectively, even for large window sizes or large values of k . Details of this algorithm are shown in Fig. 4.

Let $\eta(w)$ denote the throughput of the shorted queueing network in Fig. 1(b), provided that the population is w . Then $\eta(w)$ can be obtained using the results of the above calculation procedure and the following equation:

$$\eta(w) = \mu_N * \left(1 - \sum_{j=1}^k (P_{w,j,0} + P_{w,j,1}) \right)$$

Applying the procedure used in the Norton's theorem, the shorted queueing network shown in Fig. 1(b) is then replaced by an equivalent queue with a state-dependent service rate $\eta(w)$. Therefore, the original queueing model in Fig. 1(a) can be reduced to an approximate queueing model shown in Fig. 1(c). The queueing network shown in Fig. 1(c) is now a typical product-form queueing network. The second step of our analysis procedure is simply to solve this product-form queueing network using the approach in Ref. [22]. To summarize the above procedure, the stationary probabilities of the queueing model shown in Fig. 1(a) are obtained approximately by applying Norton's theorem for model simplification, and then the computational algorithm and product-form formulas are applied for the numerical computations.

3.2. Queueing analysis: two-threshold rate control

3.2.1. Modeling assumptions

In this section we investigate the two-threshold rate control scheme extended from the one presented above. All assumptions regarding the *Network* and *Token_holding* queues made in the previous section still hold. The only difference between these two control schemes lies in setting the threshold values in the *Network* queue. We denote the two thresholds in the control scheme of this section as L_1 and L_2 , where $L_2 \geq L_1$. The proposed method is identical to the single-threshold control scheme in last section, when $L_1 = L_2$. The case for $L_2 > L_1$ is described as follows. For convenience, in this discussion, the buffer area in the *Network* queue is segmented into three regions, I, II and III, as shown in Fig. 5(a). The *Token_holding* queue adjusts its message sending rate depending on the regions that the accumulated messages have reached in the *Network* queue (i.e. n_N). The message sending rate is set to μ_H when $n_N <$

³ Note that because the original queueing model in Fig. 1(a) is not time-reversible, we can not apply Norton's theorem to obtain an exact simplified model.

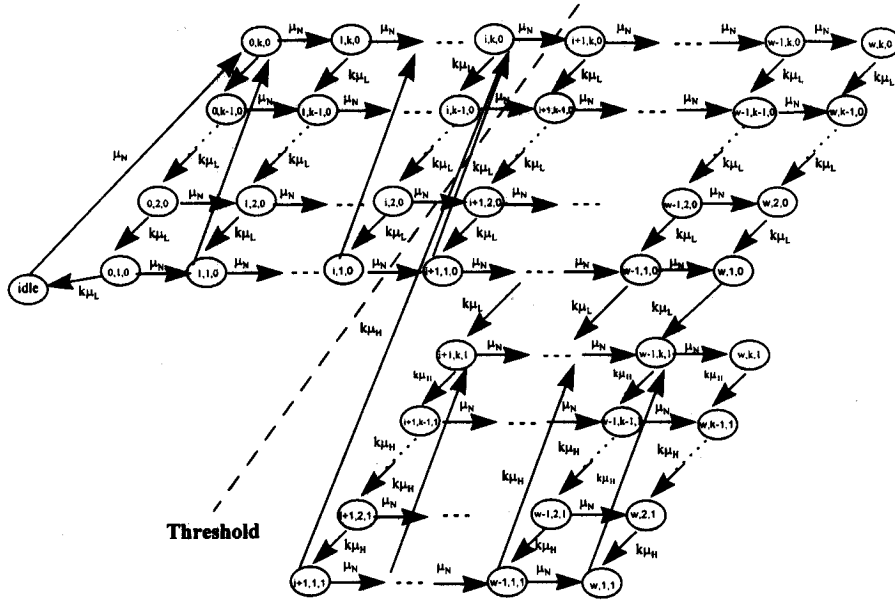


Fig. 3. State transition diagram of shorted queueing network for single-threshold rate control.

L_1 ; and to μ_L when $n_N \geq L_2$. The message sending rate remains unchanged when $L_1 \leq n_N < L_2$ (i.e. in Region II), whether it is μ_H or μ_L . Fig. 5(b) shows the sequence of message sending rate versus the number of messages in the *Network* queue. The message sending rate follows the sequence $(\mu_H, \mu_H, \text{ and } \mu_L)$ when the message queue grows continuously and reaches Regions I, II and then III. But it follows the pattern $(\mu_L, \mu_L, \text{ and } \mu_H)$ when the number of messages decreases continuously and moves from Region III to Region II, and then finally to Region I.

Based on the above descriptions, we can see that two-threshold control employs an extra buffer, Region II, to avoid frequent adjustments in the message sending rate. Implicitly, the two-threshold control scheme leads to some delay in adjusting the message sending rate when compared with the single-threshold control scheme. We now discuss what effects this buffer area may have on throughput, adjustment frequency etc., in the numerical results.

3.2.2. Steady state analysis

Again, as in Section 3.2.1, two steps are used to reduce complexity in the calculations. In step 1, we short the model at points C and D, instead of A and B. The resulting shorted queueing network is shown in Fig. 6(a). Note that Fig. 6(a) is different from Fig. 1(b) in that the *Network* queue in Fig. 1(b) is replaced with the *Ack* queue. The shorted queueing network shown in Fig. 1(b) can not be used to calculate the equivalent queueing model because the message sending rate μ_H or μ_L of the *Token_holding* queue can not be determined when n_N falls in Region II (i.e. $L_1 \leq n_N < L_2$); nor can the percentage of each rate be computed. This problem is circumvented by using the shorted queueing network in Fig. 6(a) for

approximate analysis. The steady state probability \tilde{P}_{ij} of the shorted queueing network shown in Fig. 6(a) can be calculated with a computation algorithm similar to the one proposed in previous section. Assume $\eta_{w,M}$ denotes the throughput of Fig. 6(a), where w is the total population, and M is the type of token sending rate in the *Token_holding* queue. $\eta_{w,M}$ can be computed with the equation

$$\eta_{w,M} = \mu_A * \left(1 - \sum_{j=1}^k \tilde{P}_{w,j} \right)$$

where $\tilde{P}_{w,j}$ must be solved with the sending rate μ_H if $M = 1$, and μ_L if $M = 0$.

The queueing model of Fig. 1(a) can now be simplified to that of Fig. 6(b) after step 1. The service rate $\eta_{w,M}$ of the equivalent queue from the queueing model in Fig. 6(b) is dependent on the number of messages in the *Network* queue. Since the queueing model in Fig. 6(b) is not of the BCMP type, we can only calculate the steady-state probabilities of the queueing model shown in Fig. 6(b) by numerically solving the global balance equations. Here, system state (i, j, h) stands for $n_N = i, n_R = j; M = h$. Two sets of indicators are used in the global balance equation of the system: (1) $I_{L_1}(i), \bar{I}_{L_1}(i)$, and (2) $I_{L_2}(i), \bar{I}_{L_2}(i)$, with the same definitions as $I_L(i), \bar{I}_L(i)$ in Section 3.2.1. System states $(0 \leq i \leq L_1 - 1, 1 \leq j \leq w - i, 0)$ and $(L_2 \leq i \leq w, 1 \leq j \leq w - i, 1)$ do not exist in our model, and are omitted. The global balance equations are as follows:

For $n_N = 0$,

$$\begin{cases} \eta_{w,1}P_{0,j,1} & = \mu_R P_{0,j+1,1}, & j=0 \\ (\eta_{w-j,1} + \mu_R)P_{0,j,1} & = \mu_R P_{0,j+1,1} + \mu_N P_{1,j-1,1}, & 1 \leq j \leq w-1 \\ \mu_R P_{0,j,1} & = \mu_N P_{1,j-1,1}, & j=w \end{cases}$$

Step 1: $P_{0,1,0} := 1;$

Step 2: $P_{idle} := (\frac{k\mu_L}{\mu_N})P_{0,1,0};$
 $for(j = 1; j \leq k-1; j++)$
 $P_{0,j+1,0} := (1 + \frac{\mu_N}{k\mu_L})P_{0,j,0};$

Step 3: $P_{1,1,0} := (1 + \frac{\mu_N}{k\mu_L})P_{0,k,0} - (\frac{\mu_N}{k\mu_L})P_{idle};$

Step 4: $for(i = 1; i \leq w-L; i++) \{$
 $for(j = 1; j \leq k-1; j++)$
 $P_{i,j+1,0} := (1 + \frac{\mu_N}{k\mu_L})P_{i,j,0} - (\frac{\mu_N}{k\mu_L})P_{i-1,j,0};$
 $if(i \neq w-L)$
 $P_{i+1,1,0} := (1 + \frac{\mu_N}{k\mu_L})P_{i,k,0} - (\frac{\mu_N}{k\mu_L})P_{i-1,k,0}; \}$

Step 5: $for(i = w-L+1; i \leq w; i++) \{$
 $if(i \neq w)$
 $P_{i,k,0} := (\frac{\mu_N}{k\mu_L + \mu_N})P_{i-1,k,0};$
 $else$
 $P_{i,k,0} := (\frac{\mu_N}{k\mu_L})P_{i-1,k,0};$
 $for(j = k-1; j \geq 1; j--) \{$
 $if(i \neq w)$
 $P_{i,j,0} := (\frac{\mu_N}{k\mu_L + \mu_N})P_{i-1,j,0} + (\frac{k\mu_L}{k\mu_L + \mu_N})P_{i,j+1,0};$
 $else$
 $P_{i,j,0} := (\frac{\mu_N}{k\mu_L})P_{i-1,j,0} + P_{i,j+1,0}; \}$

Step 6: $P_{w-L+1,1,1} := (\frac{k\mu_L + \mu_N}{k\mu_H})P_{w-L,k,0} - (\frac{\mu_N}{k\mu_H})P_{w-L-1,k,0} - (\frac{\mu_L}{\mu_H})P_{w-L+1,1,0};$

Step 7: $for(j = 1; j \leq k-1; j++)$
 $P_{w-L+1,j+1,1} := (1 + \frac{\mu_N}{k\mu_H})P_{w-L+1,j,1};$

Step 8: $for(i = w-L+2; i \leq w; i++) \{$
 $if(i = w-L+2)$
 $P_{i,1,1} := (1 + \frac{\mu_N}{k\mu_H})P_{i-1,k,1} - (\frac{\mu_L}{\mu_H})P_{i,1,0};$
 $else$
 $P_{i,1,1} := (1 + \frac{\mu_N}{k\mu_H})P_{i-1,k,1} - (\frac{\mu_N}{k\mu_H})P_{i-2,k,1} - (\frac{\mu_L}{\mu_H})P_{i,1,0};$
 $for(j = 1; j \leq k; j++) \{$
 $if(i \neq w)$
 $P_{i,j+1,1} := (1 + \frac{\mu_N}{k\mu_H})P_{i,j,1} - (\frac{\mu_N}{k\mu_H})P_{i-1,j,1};$
 $else$
 $P_{i,j+1,1} := P_{i,j,1} - (\frac{\mu_N}{k\mu_H})P_{i-1,j,1}; \}$

Step 9: *Normalize $P_{i,j,k}$*

Fig. 4. Computation algorithm of the shorted queueing network in single threshold rate control.

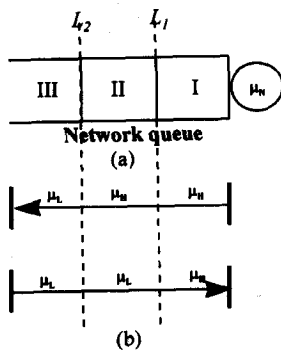


Fig. 5. (a) Three regions in the Network queue. (b) Changes in the message sending rate when the number of messages in the Network queue increases or decreases.

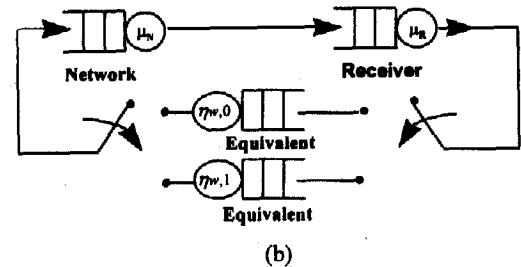
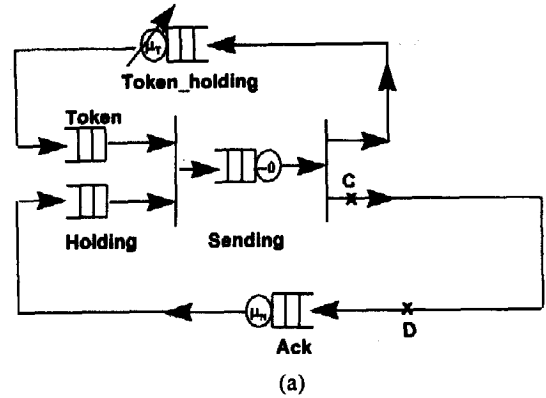


Fig. 6. (a) Shorted queueing network, and (b) equivalent queueing network.

For $1 \leq n_N < L_2,$

$$\begin{cases} (\mu_N + \eta_{w-i-1,1})P_{i,j,1} & = \eta_{w-i-j+1,1}P_{i-1,j,1} + \mu_R P_{i,j+1,1}, & j=0 \\ (\mu_R + \mu_N + \eta_{w-i-1,0})P_{i,j,1} & = \eta_{w-i-j+1,1}P_{i-1,j,1} + \bar{I}_{L_2}(i+1)\mu_N P_{i+1,j-1,1} \\ & + I_{L_1}(i+1)\mu_N P_{i+1,j-1,0} + \mu_R P_{i,j+1,1}, & 1 \leq j < w-i \\ (\mu_R + \mu_N)P_{i,j,1} & = \eta_{w-i-j+1,1}P_{i-1,j,1} + \bar{I}_{L_2}(i+1)\mu_N P_{i+1,j-1,1} \\ & + I_{L_1}(i+1)\mu_N P_{i+1,j-1,0}, & j=w-i \end{cases}$$

For $L_1 \leq n_N < w,$

$$\begin{cases} (\mu_N + \eta_{w-i-1,0})P_{i,j,0} & = \mu_R P_{i,j+1,0} + I_{L_2}(i)\eta_{w-i-j+1,1}P_{i-1,j,1} \\ & + \bar{I}_{L_1}(i)\eta_{w-i-j+1,0}P_{i-1,j,0}, & j=0 \\ (\mu_R + \mu_N + \eta_{w-i-1,0})P_{i,j,0} & = \mu_N P_{i+1,j-1,0} + I_{L_2}(i)\eta_{w-i-j+1,1}P_{i-1,j,1} \\ & + \bar{I}_{L_1}(i)\eta_{w-i-j+1,0}P_{i-1,j,0} + \mu_R P_{i,j+1,0}, & 1 \leq j < w-i \\ (\mu_R + \mu_N)P_{i,j,0} & = \mu_N P_{i+1,j-1,0} + I_{L_2}(i)\eta_{w-i-j+1,1}P_{i-1,j,1} \\ & + \bar{I}_{L_1}(i)\eta_{w-i-j+1,0}P_{i-1,j,0}, & j=w-i \end{cases}$$

For $n_N = w,$

$$\mu_N P_{w,j,0} = \eta_{1,0} P_{w-1,0,0}, \quad j=0$$

Fortunately, the total number of system states $O(W^2)$ is not too large for most rate-controlled window schemes. Hence, steady-state probabilities can be solved numerically by matrix computation. The approach proposed in this section can also be applied to analyze the rate control with a single threshold. However, the method proposed in Section 3.2.1 should be more efficient for solving their steady-state performance. Thus, we suggest that the approach presented in this section should not be used in such cases.

4. Numerical results and validation

As mentioned in the single-threshold queueing analysis in

Table 1

Percentage of relative error in average queue length of Receiver queue, with parameters $\mu_N^{-1} = \mu_A^{-1}$, $\mu_R^{-1} = 3$, $\mu_S^{-1} = 4$

μ_N^{-1}/μ_R^{-1}	$k = 1$	$k = 2$	$k = 3$	$k = 6$	$k = 10$
1	6.4%	3.0%	1.8%	0.5%	0.5%
1.5	6.4%	3.6%	2.6%	1.5%	1.5%
2	3.3%	2.0%	1.6%	1.1%	1.1%
2.5	1.6%	1.1%	0.9%	0.7%	0.7%
3	1.4%	1.1%	1.0%	0.95%	0.95%

Section 3.2.1, direct application of Norton's theorem to simplify a queueing model that is not time-reversible leads only to an approximate model. We verify the applicability of such an approximation by simulation. The corresponding results are shown in Tables 1 and 2, from which we conclude that a timer of constant value can be approximated in our model using an Erlang- k random variable with large k , $k \geq 2$. Next, we investigate other statistical characteristics of the network under the rate-controlled window scheme.

In the first experiment, we investigate the effect of message rate reduction on network performance and end-system performance by gradually adjusting the message sending rate, (μ_T). We apply the same rate for μ_H and μ_L in this experiment and call it the single rate control (SRC) approach. Three different window sizes (i.e. $W = 4, 8, 12$) are chosen in this experiment. Results are then compared with those of a traditional sliding window flow control scheme operating with the same set of window sizes. The local environment assumes the one-way distance to be no more than 5 km. With OC-3c or STM-1 as the physical media for this experiment, the worst-case round-trip delay is equal to $28.4 \mu s$ (≈ 10 time slots). Fig. 7 shows that the average queue length in the receiver entity (Receiver queue) is much shorter when rate control is employed. The mean queue length decreases to a considerably lower level when the service rate of the receiver entity is larger than twice the message sending rate. Fig. 8 shows that the second moment of queue length in the Receiver is also much shorter under such a rate-controlled window scheme. Finally, the average queue length of the Network is plotted in Fig. 9.

From the above observations, we have verified that rate control is more effective in reducing congestion at the Receiver, as well as the ATMLAN, than traditional window flow control. However, as shown in Fig. 10, the rate control

scheme may cause longer mean network cycle times than window control and leads to lower throughput. This trade-off seems unavoidable. For comparison, our models are simulated with the worst-case propagation delay (10 slots) or negligible delay. From Figs. 7, 8 and 10, we can justify that the end-to-end propagation delay is negligible in a local environment.

We illustrate the power of rate control in the second experiment, using the single-threshold rate control (STRC) with $\mu_H \neq \mu_L$. In addition, the effect of non-zero feedback delay is verified in STRC approach. In this experiment, we also assume the round-trip delay is equal to $28.4 \mu s$. We set $\mu_S^{-1} = 30$ slots for the SRC, $\mu_H^{-1} = \mu_S^{-1}$, $L = 3$ and $W = 8$. The message sending rate is reduced to one-eighth of the original rate (i.e. $\mu_L = 1/8\mu_H$) when the queue length in the network node reaches the threshold. Three curves are depicted. Two curves of SRC are obtained with the message sending rate set equal to μ_H and μ_L . The other one illustrates the results of the STRC. In Fig. 11 we find that the average queue length of the Receiver is smaller in STRC than that in SRC. In our STRC, we adjust the message sending rate on the sending entity according to the queue length of the Network. Thus, the message population of the Network is of more interest. As one might expect, we can observe from Fig. 12 that the average queue length of the Network is much shorter when the STRC is employed than when it is under SRC. From Fig. 13, it is clear that STRC does reduce the second moment of the queue size in the Network. Fig. 14 shows that the mean cycle time in the dynamic rate control mode is close to that of the SRC. Therefore, using STRC is still a better choice due to the fact that its throughput can be adjusted to be close to that of SRC while the congestion or buffer overflow in the switch can be successfully avoided. From these experiments, we can conclude that the results of our simulations coincide well with the analytical results.

Table 2

Percentage of relative error in mean network cycle time, with parameters $\mu_N^{-1} = \mu_A^{-1}$, $\mu_R^{-1} = 3$, $\mu_S^{-1} = 4$

μ_N^{-1}/μ_R^{-1}	$k = 1$	$k = 2$	$k = 3$	$k = 6$	$k = 10$
1	4.9%	3.2%	2.7%	2.2%	2.0%
1.5	4.7%	3.2%	2.7%	2.1%	1.9%
2	2.1%	1.4%	1.2%	0.96%	0.87%
2.5	1.25%	0.9%	0.8%	0.7%	0.64%
3	0.6%	0.43%	0.37%	0.32%	0.29%

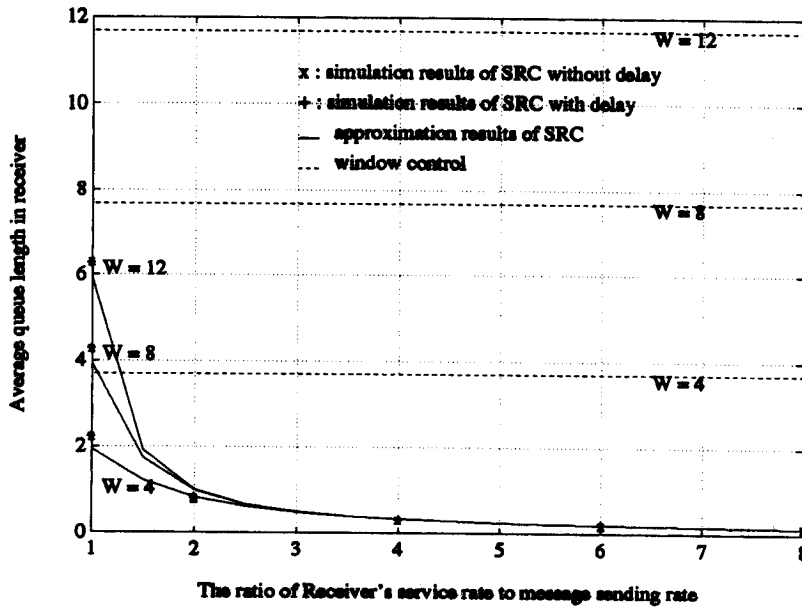


Fig. 7. μ_R/μ_H versus average queue length in Receiver, using OC-3c interface with parameters: VC capacity $\mu_N = 150$ Mbps, $\mu_A = \mu_N$, $\mu_R = 20$ Mbps, round-trip delay = 10 slots, mean packet size ~ 2120 bytes and $k = 2$.

The assumption of zero feedback delay in our model is again found to be reasonable from Figs 11, 12, and 14.

Next, we compare the performance of STRC with that of two-threshold rate control (2TRC). We observe the following performance and congestion metrics: throughput, mean and second moment of the Network queue length, which are used as metrics of the network congestion level, and the adjustment frequency of the message rate. A single scenario is considered and the physical media is assumed to be OC-12 or STM-4 (600 Mbps data rate). Because the

propagation delay is justified to be negligible in the above experiments, we assume zero propagation delay in the following scenario.

In this scenario, we change the granted bandwidth for the connecting virtual channel (VC) according to the following selected parameters: $\mu_R = 250$ Mbps, $\mu_H^{-1} = 200$ slots, $\mu_L^{-1} = 2000$ slots, $k = 5$, $W = 16$ and $\mu_A = 1.1 \mu_N$. The latter parameters imply that the forward direction in the ATM LAN is more seriously congested than the backward direction and thus requires dynamic rate control. To ensure a fair

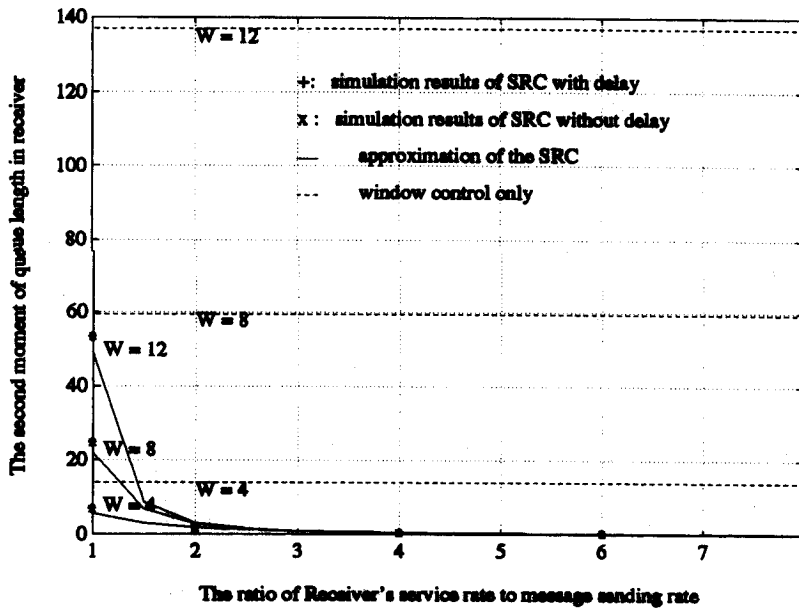


Fig. 8. μ_R/μ_H versus the second moment of queue length in Receiver, using OC-3c interface with parameters: VC capacity $\mu_N = 150$ Mbps, $\mu_A = \mu_N$, $\mu_R = 20$ Mbps, round-trip delay = 10 slots, mean packet size ~ 2120 bytes and $k = 2$.

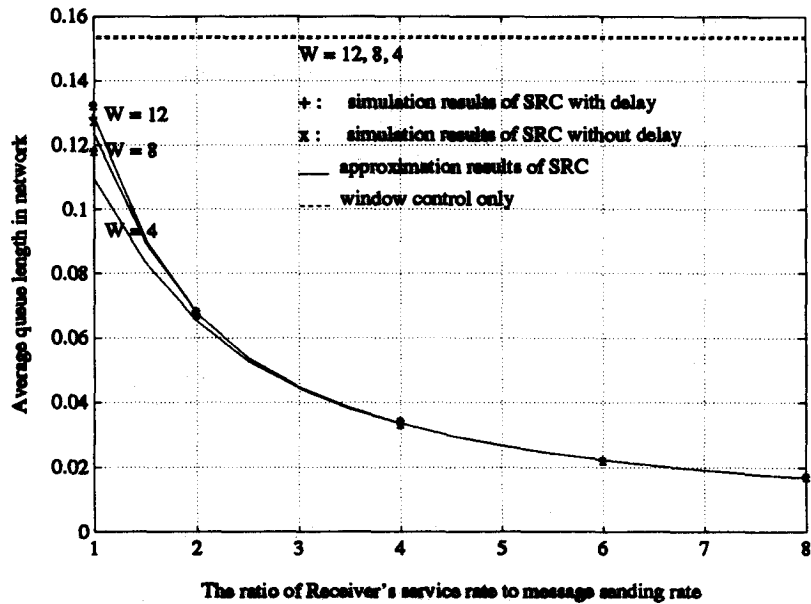


Fig. 9. μ_R/μ_H versus the average queue length in Network, using OC-3c interface with parameters: VC capacity $\mu_N = 150$ Mbps, $\mu_A = \mu_N$, $\mu_R = 20$ Mbps, round-trip delay = 10 slots, mean packet size ~ 2120 bytes and $k = 2$.

comparison of STRC and 2TRC, we select $L = 8$ for the STRC and (8,12) (6,10) (4,8) (2,8) for (L_1, L_2) in 2TRC. Note that we have kept the gap between L_1 and L_2 constant for (8,12) (6,10) (4,8), while the case (2,8) is used to illustrate the impact of gap size on system performance. Fig. 15 shows that the throughput increases monotonically as the virtual channel capacity (Network service rate μ_N) increases under both schemes. Actually the throughputs for these selected cases are so close that no simple conclusion can be drawn as to which control policy is superior. However, when the adjustment frequency is

considered, 2TRC usually yields superior performance, as shown in Fig. 16. Only when the VC bandwidth is much higher or much lower than the message sending rate operating region, can STRC be found to have similar adjustment overhead. Meanwhile, as shown in Figs 17 and 18, congestion conditions in the Network are found to be more effectively alleviated by 2TRC. The cost of such improvement in congestion control is actually fairly limited. For example, setting $L_1 = 4$ and $L_2 = 8$ could lead to a lower mean queue length in the Network (see Figs 17 and 18), but very limited throughput degradation and adjustment overhead are found.

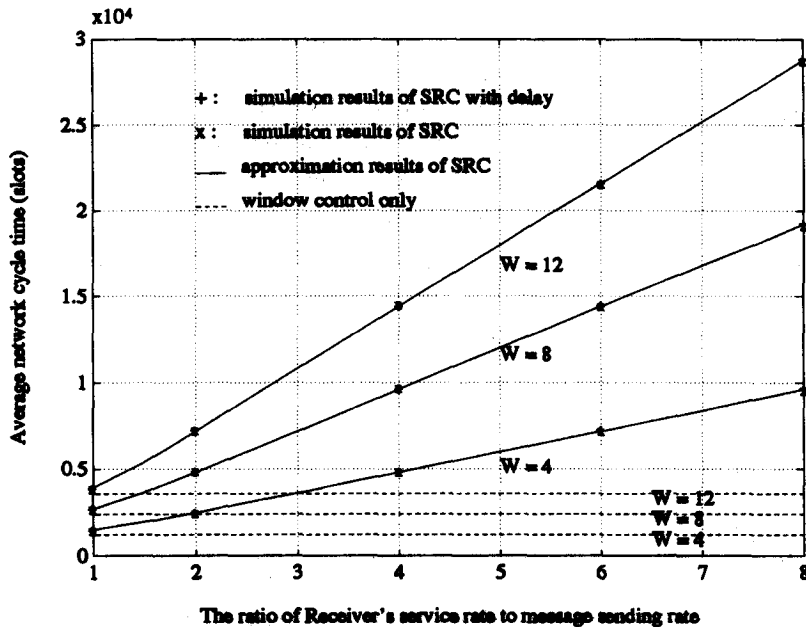


Fig. 10. μ_R/μ_H versus average network cycle time, using OC-3c interface with parameters: VC capacity $\mu_N = 150$ Mbps, $\mu_A = \mu_N$, $\mu_R = 20$ Mbps, round-trip delay = 10 slots, mean packet size ~ 2120 bytes and $k = 2$.

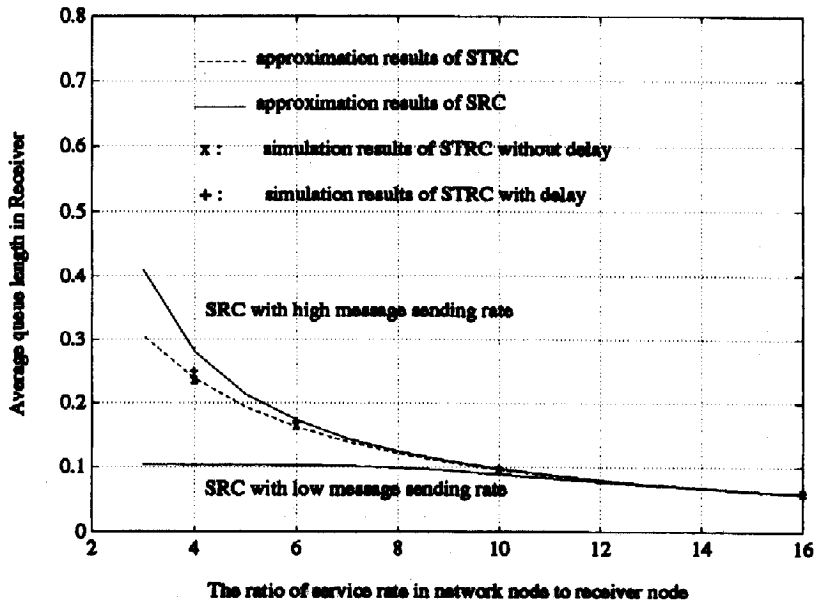


Fig. 11. μ_R/μ_N versus average queue length in Receiver, using OC-3c interface with parameters: $\mu_R = 320$ Mbps, $\mu_A = \mu_N$, $\mu_L^{-1} = 8\mu_H^{-1}$, $\mu_H^{-1} = 30$ slots, round-trip delay = 10 slots, mean packet size ~ 3400 bytes, $W = 8$, $L = 3$, and $k = 2$.

When the VC capacity approaches zero, the mean queue length in the Network (see Fig. 17) coincides with the results predicted by the BCMP product-form solution when $\mu_A = 1.1\mu_N$.

To investigate the impact of threshold values on the system performance, we also tested different gap sizes between high and low thresholds. Threshold L_2 is set to 8, while L_1 is set to 2 or 4 (i.e. $(L_1, L_2) = (2, 8), (4, 8)$). It can be observed from Fig. 15 that STRC and 2TRC with small gap sizes actually yield higher throughput. This result is not surprising since the message sending rate

is kept at the higher rate more frequently. The cost of higher throughput corresponds to a higher adjustment frequency, as shown in Fig. 16, and a higher mean and second moment of the queue length in the Network, as shown in Figs 17 and 18.

5. Conclusions

In this paper we have investigated the performance of a generic rate-controlled window scheme that can be

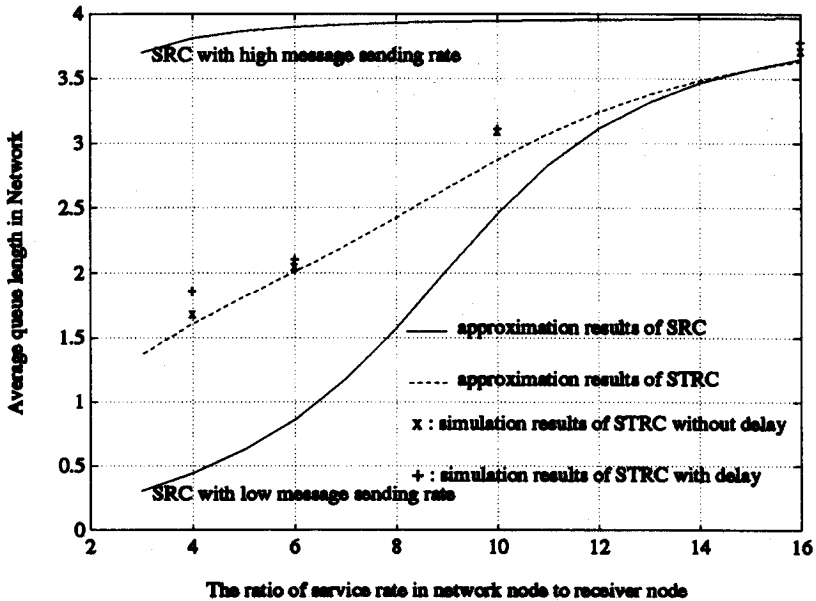


Fig. 12. μ_R/μ_N versus average queue length in Network, using OC-3c interface with parameters: $\mu_R = 320$ Mbps, $\mu_A = \mu_N$, $\mu_L^{-1} = 8\mu_H^{-1}$, $\mu_H^{-1} = 30$ slots, round-trip delay = 10 slots, mean packet size ~ 3400 bytes, $W = 8$, $L = 3$, and $k = 2$.

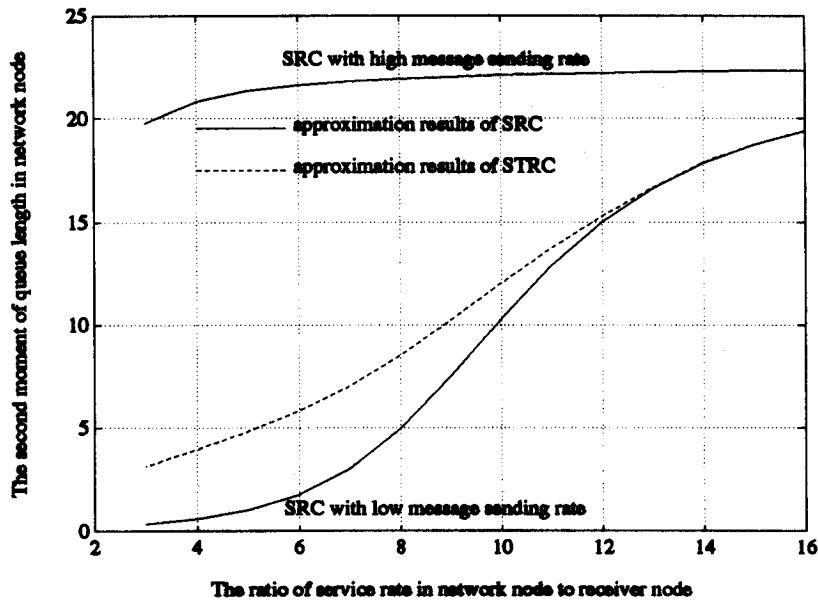


Fig. 13. μ_R/μ_N versus average queue length in Network, using OC-3c interface with parameters: $\mu_R = 320$ Mbps, $\mu_A = \mu_N$, $\mu_L^{-1} = 8\mu_H^{-1}$, $\mu_H^{-1} = 30$ slots, round-trip delay = 10 slots, mean packet size ~ 3400 bytes, $W = 8$, $L = 3$, and $k = 2$.

employed in transport protocols for ATM LAN and other high-speed network environments. The steady-state performance is obtained using Norton's approximation in closed queueing network models. Our equivalent queue approach, Erlang- k and zero-propagation delay model assumptions have significantly reduced the complexity in queueing analysis, and yet yield highly accurate results, as illustrated in the numerical examples.

In the numerical results, the rate control schemes have been shown to have significant effects in reducing

congestion levels at the receiving entity as well as in the LAN switches. When the performances of the single-rate and single-threshold rate control policies are compared, single-threshold rate control is found to provide nearly the same throughput as the single-rate rate control operating under a high sending rate, while keeping the message population in the receiver node or the switches at a relatively low level.

We also conclude that the two-threshold rate control scheme can be more effective in alleviating network

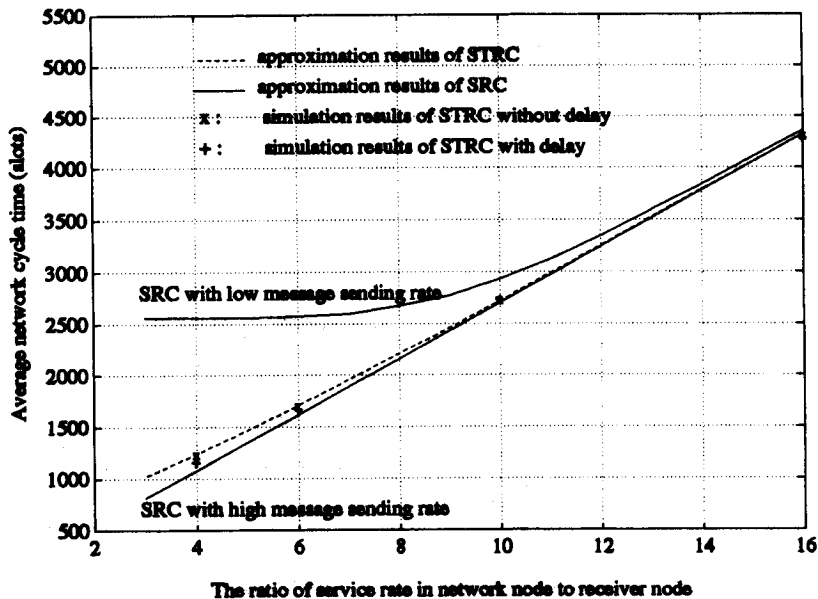


Fig. 14. μ_R/μ_N versus average network cycle time, using OC-3c interface with parameters: $\mu_R = 320$ Mbps, $\mu_A = \mu_N$, $\mu_L^{-1} = 8\mu_H^{-1}$, $\mu_H^{-1} = 30$ slots, round-trip delay = 10 slots, mean packet size ~ 3400 bytes, $W = 8$, $L = 3$, and $k = 2$.

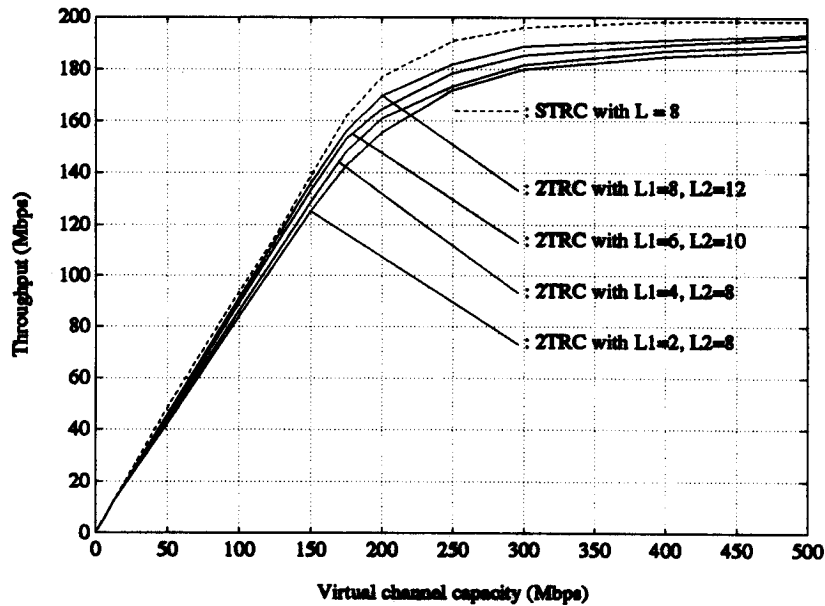


Fig. 15. VC capacity μ_N versus throughput, using OC-12 interface with parameters: $\mu_A = 1.1 \mu_N$, $\mu_R = 250$ Mbps, $\mu_H^{-1} = 200$ slots, $\mu_L^{-1} = 2000$ slots, round-trip delay = 40 slots, mean packet size ~ 3540 bytes, $W = 16$, and $k = 5$.

congestion than the single-threshold scheme, while keeping the control overhead (in terms of the rate adjustment frequency) at a low level, provided that both high and low threshold values are selected appropriately. The total throughput could be affected, but only in a limited fashion. In addition, one should not consider the throughput as the only objective. Where network congestion is not concerned at all, this will only lead to the trivial conclusion that rate control is not desired and the message sending rate should be always be kept high.

Currently in our model, both message losses and the feedback delay are assumed to be negligible. The latter assumption has been found to be reasonable for the local area ATM network in our numerical example, mainly due to the fact that the non-preemptive nature of the message-based rate control mechanism can absorb most propagation delay. However, how to develop new analytical models that reflect the message loss behavior under network congestion and non-negligible delay remains an area that requires further investigation.

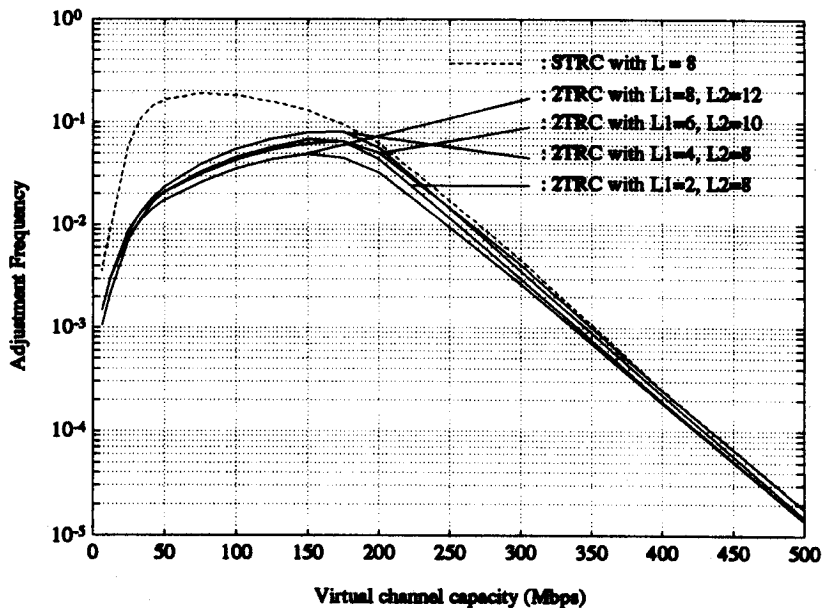


Fig. 16. VC capacity μ_N versus adjustment frequency, using OC-12 interface with parameters: $\mu_A = 1.1 \mu_N$, $\mu_R = 250$ Mbps, $\mu_H^{-1} = 200$ slots, $\mu_L^{-1} = 2000$ slots, round-trip delay = 40 slots, mean packet size ~ 3540 bytes, $W = 16$, and $k = 5$.

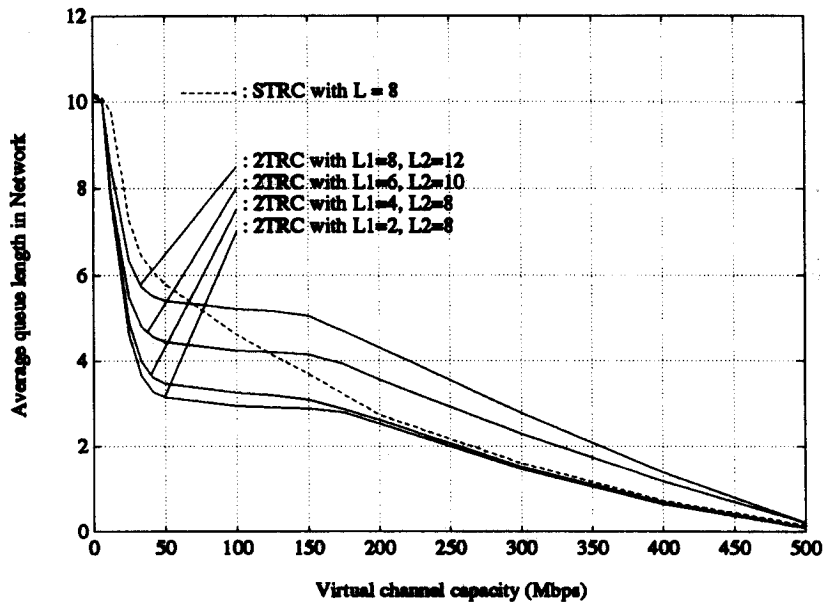


Fig. 17. VC capacity μ_N versus average queue length in Network, with parameters using OC-12 interface with parameters: $\mu_A = 1.1 \mu_N$, $\mu_R = 250$ Mbps, $\mu_H^{-1} = 200$ slots, $\mu_L^{-1} = 2000$ slots, round-trip delay = 40 slots, mean packet size ~ 3540 bytes, $W = 16$, and $k = 5$.

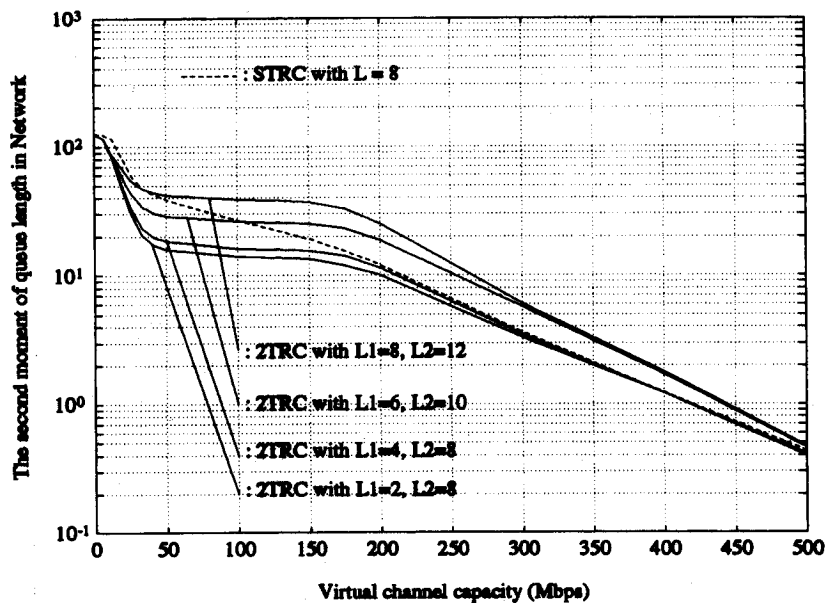
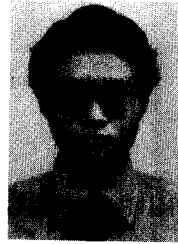


Fig. 18. VC capacity μ_N versus the second moment of queue length in Network, using OC-12 interface with parameters: $\mu_A = 1.1 \mu_N$, $\mu_R = 250$ Mbps, $\mu_H^{-1} = 200$ slots, $\mu_L^{-1} = 2000$ slots, round-trip delay = 40 slots, mean packet size ~ 3540 bytes, $W = 16$, and $k = 5$.

References

- [1] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison-Wesley, Reading, MA, 1988.
- [2] T. Toniatti, F. Trombetta, Performance simulation of end-to-end windowing in ATM networks, IEEE INFOCOM '92, 1992, pp. 495–502.
- [3] D. Mitra, Asymptotically optimal design of congestion control for high-speed data networks, IEEE Trans. Commun. 40 (1992) 301–311.
- [4] H. Kwon, A. Tubtiang, G. Pujolle, A simple flow control mechanism in ATM networks with end-to-end transport, IEEE INFOCOM '93, 1993, pp. 654–661.
- [5] A.K. Agrawala, D. Sanghi, Design and evaluation of an adaptive flow control scheme, IEEE INFOCOM '92, 1992, pp. 2391–2397.
- [6] J. Bolot, A.U. Shankar, Dynamic behavior of rate-based flow control mechanisms, ACM SIGCOMM '90, 1990, pp. 35–49.
- [7] L. Benmohamed, S.M. Meerkov, Feedback control of congestion in packet switching networks: the case of a single congested node, IEEE/ACM Trans. Networking 1 (1993) 693–708.

- [8] D. Cheriton, VMTP: Versatile message transaction protocol: protocol specification, RFC 1045, February 1988.
- [9] W.T. Strayer, B.J. Dempsey, A.C. Weaver, XTP: The Xpress Transfer Protocol, Addison-Wesley, Reading, MA, 1992.
- [10] D. Clark, M.L. Lambert, L. Zhang, NETBLT: A Bulk Data Transfer Protocol, RFC 998, March 1987.
- [11] A.E. Conway, A perspective on the analytical performance evaluation of multilayered communication protocol architectures, *IEEE J. Select. Areas Commun.* 9 (1991) 4–14.
- [12] L.C. Mitchell, D.A. Lide, End-to-end performance modeling of local area networks, *IEEE J. Select. Areas Commun.* 4 (1986) 975–985.
- [13] M. Murata, H. Tagaki, Two-layer modeling for local area networks, *IEEE Trans. Commun.* 36 (1988) 1022–1034.
- [14] S. Fdida, H.G. Perros, A. Wilk, Semaphore queues: modeling multilayered window flow control mechanisms, *IEEE Trans. Commun.* 38 (1990) 309–317.
- [15] G.W. Shapiro, H.G. Perros, Nested sliding window protocols with packet fragmentation, *IEEE Trans. Commun.* 41 (1993) 99–109.
- [16] A. Romanow, S. Floyd, Dynamics of TCP traffic over ATM networks, *IEEE Trans. Commun.* 13 (1995) 633–641.
- [17] W.A. Doeringer, D. Dykeman, M. Kaiserswerth, B.W. Meister, H. Rudin, R. Williamson, A survey of light-weight transport protocols for high-speed networks, *IEEE Trans. Commun.* 38 (1990) 2025–2038.
- [18] D.W. Browning, Flow control in high-speed communication networks, *IEEE Trans. Commun.* 42 (1994) 2480–2489.
- [19] Y.T. Wang, B. Sengupta, Performance analysis of a feedback congestion control policy under non-negligible propagation delay, *ACM SIGCOMM '91*, 1991, pp. 149–157.
- [20] K.M. Chandy, U. Herzog, L. Woo, Parametric analysis of queueing networks, *IBM J. Res. Dev.* 19 (1975) 36–42.
- [21] P.S. Kritzinger, S. Van Wyk, A.E. Krzesinski, A generalization of Norton's theorem for multiclass queueing networks, *Performance Evaluation* 2 (1982) 98–107.
- [22] F. Baskett, K.M. Chandy, R.R. Muntz, F.G. Palacios, Open, closed and mixed networks of queues with different classes of customers, *J. ACM* 22(2) (1975) 248–260.



Hong-Bin Chiou received his B.S. degree (1986) from National Taiwan University and M.S. degree (1989) from National Tsing Hwa University, both in electrical engineering. In 1989–91 he worked as a project researcher in the PC System Design Department of the R&D Division of Acer, Inc. In 1991 he joined the Telecommunication Laboratories of the Ministry of Transportation and Communications (MOTC), Taiwan, and is involved in research and development of multimedia communication systems. He is currently in the PhD program in the National Taiwan University and is interested in flow control and buffer management schemes for high-speed networks. Mr Chiou is also a member of Phi Tau Phi.



Zsehong Tsai received his B.S. degree (1983) in electrical engineering from National Taiwan University, Taipei, and M.S. (1985) and PhD degrees (1988) from the University of California, Los Angeles. In 1988–90 he worked as a member of technical staff at AT&T Bell Laboratories, where he investigated performance issues related to network management systems. Since 1990 he has been an Associate Professor in the Department of Electrical Engineering of National Taiwan University. Currently he is also with the Institute of Industrial Engineering and the Center of Telecommunication Research at National Taiwan University. His interests include high-speed networking, wireless communications, network management, and performance of telecommunication networks.