



ELSEVIER

Available at  
www.ComputerScienceWeb.com  
POWERED BY SCIENCE @ DIRECT®

Computer Standards & Interfaces 25 (2003) 515–528

COMPUTER STANDARDS  
& INTERFACES

www.elsevier.com/locate/csi

## Ping-pong flow control for ATM ABR traffic

Cheng-Yuan Ku<sup>a,\*</sup>, Lain-Chyr Hwang<sup>b</sup>, Hsin-Tzung Wu<sup>c</sup>

<sup>a</sup>Department of Information Management, National Chung Cheng University, 160 San-Hsing, Ming-Hsiung, ChiaYi, Taiwan, ROC

<sup>b</sup>Department of Electrical Engineering, I-Shou University, Taiwan, ROC

<sup>c</sup>Department of Communication Engineering, National Chiao Tung University, Taiwan, ROC

Received 23 September 2002; received in revised form 23 September 2002; accepted 19 January 2003

### Abstract

In this paper, we propose an improved technique for congestion control, named as ping-pong flow control (PPFC), for asynchronous transfer mode (ATM) available bit rate (ABR) traffic. This is a rate-based flow control scheme, in which the rate regulation is achieved by directly adjusting the transmission rate in the source end station. The proposed algorithm uses a bipolar feedback strategy, which employs positive and negative feedbacks to control the transmission rate for different switch states. These states are determined using the traditional threshold-based method. We also introduce state early detection (SED), which enables the PPFC to control traffic flows more precisely and accurately at critical moments. The simulation results show that the proposed algorithm provides a higher throughput and lower cell loss ratio when compared to the well-known backward explicit congestion notification (BECN). Furthermore, these results also show that PPFC is robust against feedback losses.

© 2003 Elsevier Science B.V. All rights reserved.

*Keywords:* ATM; ABR; QoS; Ping-pong flow control; Bipolar feedback; State early detection

### 1. Introduction

The major characteristics of the asynchronous transfer mode (ATM) networking system are its data transmission and bandwidth management capacity. In general, the ATM forum defined the following five service classes to support different traffic types: constant bit rate (CBR), real-time

variable bit rate (rt-VBR), non-real-time variable bit rate (nrt-VBR), available bit rate (ABR), and unspecified bit rate (UBR). The CBR is often utilized in circuit emulation, since it provides a stable transmission environment. Similarly, VBR is frequently used to carry compressed multimedia data, because the variable bandwidth and bursty tolerance fits the requirements for compressed multimedia data.

The main goal of ABR is to use the remaining bandwidth left unused by CBR and VBR to transmit data in a best effort manner. The users first declare a set of parameters including the peak cell rate (PCR) and minimal cell rate (MCR) while the call

\* Corresponding author. Tel.: +886-5-2720411x34603; fax: +886-5-2721501.

E-mail addresses: cooperku@mis.ccu.edu.tw (C.-Y. Ku), lain@isu.edu.tw (L.-C. Hwang), sjon@isu.edu.tw (H.-T. Wu).

is setup. Feedback flow control algorithms are adopted to regulate the transmission rate of the ABR connection between the PCR and MCR. Basically, ATM flow control algorithms can be classified into two categories according to the control mechanisms: the credit-based flow control [1] and the rate-based flow control [2,3]. Credit-based flow control usually uses a hop-to-hop feedback control loop to regulate the traffic flow between two neighboring nodes. Instead of using a hop-to-hop control loop, the rate-based flow control employs an end-to-end feedback control loop to manage traffic flow between the source (SES) and destination end stations (DES), which are linked via one or more intermediate switches. Rate regulation is achieved by directly adjusting the transmission rate of the SES. ATM Forum chose the rate-based flow control in 1995 [4].

Rate-based control algorithms can be further classified into explicit indication (EI) scheme and explicit rate (ER) schemes, depending on the rate adjustment techniques used in SES. In the EI scheme, the allowed cell rate (ACR) in SES is increased or decreased using linear, exponential or some other rules by considering the network traffic. The well-known FECN, backward explicit congestion notification (BECN), and PRCA are EI-based schemes [5–10]. However, the ER scheme directly adjusts the ACR to a suitable value. The EPRCA, APRCA, OPRCA, and ERAMM all use such a scheme [11–14]. The ER scheme converges quickly, but it will increase the complexity of the switch design to support greater computation. Conversely, the hardware cost for the EI-based switch is much lower, since the control algorithm is simpler. This leads EI schemes to keep flexibility and extensibility in the development of the control algorithm. If the rate regulation is well designed, ACR convergence in EI schemes will be improved and may be even better than ER schemes. This is because the available bandwidth changes constantly.

In this paper, we propose an improved EI-based flow control algorithm for ABR, which employs a hybrid feedback strategy to control the transmission rate of the traffic source. According to the simulation results, our algorithm has the advantages of efficient rate regulation and insensitivity to feedback delay and loss. These benefits not only reduce cell loss ratio, but

also keep the switch queue length within a desired range. We organized this paper as follows. The framework of the ABR flow control is described in Section 2. Our algorithm and its implementation is proposed in Section 3 and the simulation results are presented in Section 4. Finally, we conclude this paper in Section 5.

## 2. Framework of ABR flow control

The objective of ABR flow control is to fully utilize the remaining available bandwidth and avoid cell losses, which usually occur while a switch is out of the buffer. In most cases, these control algorithms should make decisions on the status of intermediate switches. A good flow control algorithm should prevent cell losses by properly reducing the traffic flow before buffer space is exhausted and keep the bandwidth fully utilized. Based on FECN, BECN, and PRCA, we propose an improved ping-pong flow control (PPFC) algorithm, which is an EI-based flow control scheme. In general, an EI-based flow control algorithm is implemented using mechanisms with three basic units: state detection, feedback strategy, and rate regulation. The state detection technique is used to recognize the congestion states of an intermediate switch. The control mechanism would then control the traffic flow of the SES based on this information. Most algorithms use the threshold to determine the states of a switch. However, threshold selection would be another important issue for the performance of the flow control algorithm [15]. As for the feedback strategy, it will define how the switch information is represented using indication bits sent back to the SES. If the reaction for receiving the indication bit is to increase the transmission rate, then the feedback scheme is positive, as adopted in PRCA. Similarly, if the reaction is to decrease the transmission rate, then it is a negative feedback, as used in FECN. Rate regulation is a mechanism to decide how the transmission rate is adjusted. The exponential and linear schemes are the two schemes often adopted. The linear scheme is conservative but stable, whereas the exponential scheme is aggressive but variable. How to design a suitable regulation is a critical task for system performance.

### 3. Ping-pong flow control

#### 3.1. Ping-pong flow control algorithm

Consider a simplified system with one SES, one DES, one virtual circuit (VC) between the SES and DES and in-between switches on this VC, as shown in Fig. 1. State detection is an important issue in flow control design. In principle, it determines whether a switch is congested or idle. The source transmission rate is then decreased or increased, since the rate regulation is triggered by the change in switch states. Unlike most of the other algorithms, which classify the system status into two categories, PPFC defines triple states. We define a switch to be idle if its buffer is nearly empty. The switch utilization of the output link is then poor. If all switches in a system are idle, then this system is underloaded. The source transmission rate should be increased using a positive feedback forward control loop. A switch is congested if its buffer is nearly full. Although the utilization of its output link is good, free buffers may exhaust soon and it will result in cell loss in the near future. A system is overloaded if there is at least one congested switch inside. PPFC would try to recover using a negative feedback backward control loop to quickly decrease the source transmission rate. A switch is defined as stable if its buffer is not nearly empty or full. A switch in this “best area” holds enough buffers and always has cells to send [15]. To keep all the switches in a system running in such a best area is a major concern for any flow control algorithm. Hence, PPFC defines two queue length thresholds, which determines if the switch is idle, stable, or congested. However, as we observe from many experimental results, the variation of queue length is quite wavy. Therefore, single-threshold-type separation will cause the switch to change status too frequently while approaching this threshold. We then define threshold

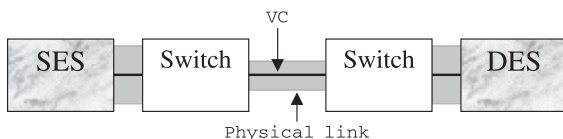


Fig. 1. System and its components.

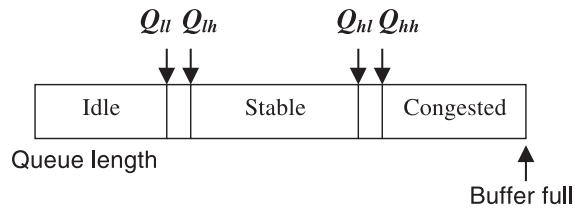


Fig. 2. Two threshold bands and corresponding switch state.

bands  $Q_h = \{Q_{hh}, Q_{hl}\}$  and  $Q_l = \{Q_{lh}, Q_{ll}\}$ , which denote the higher and lower sets of thresholds, respectively. The values of these thresholds always have the order of  $Q_{hh} > Q_{hl} > Q_{lh} > Q_{ll}$  as shown in Fig. 2. While the original switch state is idle and the queue length is getting larger than  $Q_{lh}$ , then change the switch state as stable. While the original switch state is stable and the queue length is getting smaller than  $Q_{ll}$ , then change switch state as idle. By similar way, while the original switch state is stable and the queue length is getting larger than  $Q_{hh}$ , then change the switch state as congested. While the original switch state is congested and the queue length is getting smaller than  $Q_{hl}$ , then change the switch state as stable. In this way, the wavy variation of queue length will not change the switch status too frequently. Of course, the choice of these values significantly influences the system performance. Lai and Lin [15] suggested a good way to choose these thresholds, however, they still considered only double states.

Although switch states can be determined simply by comparing the current queue length and thresholds, longer transmission delay, i.e., propagation delay and queuing delay could cause SES not to respond immediately. This will result in a higher cell loss rate, lower utilization, and more demand for the buffer space. Therefore, we designed a simple technique, namely, state early detection (SED) to reduce the impact of various network delays. SED uses the variation in queue length to gain the information of the current tendency of the switch states. A switch, which supports SED, would periodically record its current queue length and the last queue length. We then define:

$$\Delta Q(i) = Q(i) - Q(i - t).$$

Here,  $Q(i)$  denotes the queue length in time  $i$  and  $t$  denotes a small time interval. If  $\Delta Q(i)$  is positive, then the input rate is larger than the output rate during most of the past time period  $t$ . Even if a switch is idle, since its queue length seems to be growing, it will eventually leave the idle state and the action of increasing rate will be stopped soon. Conversely, if  $\Delta Q(i)$  is negative, the input rate is smaller than the output rate during most of the past time interval  $t$ . Even though a switch might be congested now, since the queue length seems to be shrinking, the congested situation will finally be relieved and the action of decreasing rate will be stopped soon. As well known, it is difficult to predict the tendency of the switch states, since  $Q(i)$  would change rapidly within a short moment in high-speed networks. Thus, switches have to also react rapidly to keep its buffer from being emptied or overloaded. To achieve this goal, the control algorithm needs to know the tendency of  $Q(i)$  within the next time period,  $t$ , to have proper reactions. SED is just designed to predict  $Q(i+t)$  by  $Q(i)$  and  $\Delta Q(i)$ . Because the time period  $t$  is very small, it can roughly predict the tendency of  $Q(i)$  within the next time period  $t$ . According to the results of SED, if there is a large probability that the switch state will change then an early action will substantially reduce the variation of queue length. In Ref. [16], an exponentially weighted average queue length is employed to predict a switch state. Actually, much earlier information has little influence on the tendency of switch state while the network environment is changing rapidly. So, a simple SED is good enough for the control mechanism of ATM ABR traffic to act in advance.

For the control mechanism, PPFC uses both positive and negative feedbacks, named as bipolar feedback, to control the source transmission rate.

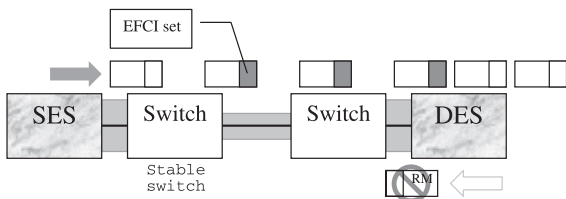


Fig. 3. Operation for stable system.

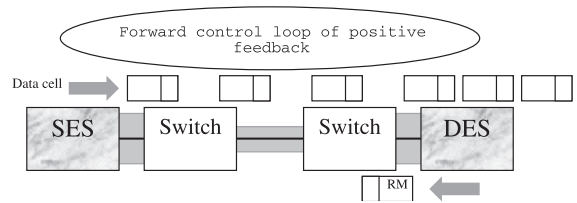


Fig. 4. Operation for underloaded system.

The source transmission rate will remain unchanged without receiving any feedback. The positive feedback on the control loop, which extends forward from SES to DES and then back to SES, will increase the transmission rate. The negative feedback on the control loop, which spans backward from a congested node to SES, will decrease the transmission rate. Cells are originally transmitted from SES with the EFCI bit cleared. Intermediate switches will mark this bit, if they are non-idle, or going to leave this state soon, i.e.  $\Delta Q(i) > T_i$ , where  $T_i$  is a positive integer, called increment threshold, which is used to indicate the strength of queue growing. DES would periodically check the EFCI bit for the cells received in this cycle. If any one of the received EFCI bit was marked, DES will not initiate any further positive feedback and the transmission rate will not increase, since it was notified that the system is not underloaded or going to leave this state, as shown in Fig. 3. Conversely, DES recognizes that the system is underloaded. It then generates a positive feedback to SES in the form of an ATM RM cell with  $CI=NI=0$ . The SES increases its transmission rate upon receiving the RM cell with  $CI=NI=0$  if the system state is underloaded, as shown in Fig. 4. This control loop is the forward positive feedback. Moreover, a congested switch not only continuously marks EFCI bits on passing cells to restrain positive feedback, but also periodically generates negative feedback by sending an RM cell with  $CI=NI=1$  back to the SES until the congestion is relieved or going to be relieved, that is,  $\Delta Q(i) < (-T_d)$ , where  $T_d$  is also a positive integer called decrement threshold, used to indicate the strength of queue shrinking. The SES immediately decreases its transmission rate upon receiving the RM cell with  $CI=NI=1$ , as shown

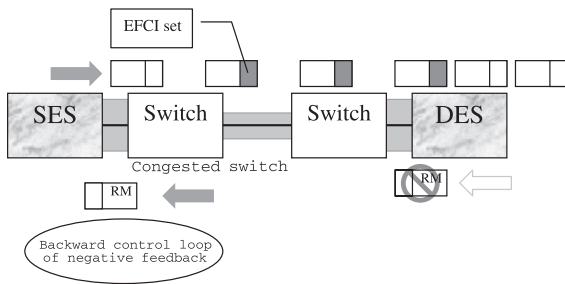


Fig. 5. Operation for overloaded system.

in Fig. 5. In PPFC, we adopt a linear-based rate regulation mechanism that can fairly adjust transmission rate rapidly in most network environments. Traditional exponential rate regulation changes the transmission rate very rapidly, and it thus works well only in the network environment with large variation of flow rates. It can be even worse while it is used in the network with a larger delay-bandwidth product [16], such as a wide area network (WAN) or high-speed network. So, in PPFC, the new ACR is computed as follows:

$$ACR = \min(ACR + (PCR - MCR) \times K\%, PCR)$$

if it needs to be increased

$$ACR = \max(ACR - (PCR - MCR) \times K\%, MCR)$$

if it needs to be decreased

where  $K$  is the multiplicative ratio of the difference between PCR and MCR. The choice of larger  $K$  works well in the network environment with huge variation of flow rates. For mild environment, the network administrator can choose the ratio to be smaller. Also notice that, SES is not confused by the positive and negative feedbacks, since the CI bit and NI bit can be used to identify the type of feedback. Additionally, switches send non-idle signal to SES when their states change from idle to non-idle. The signal, carried by RM cell with CI=0 and NI=1, is used to notify that the switch is no

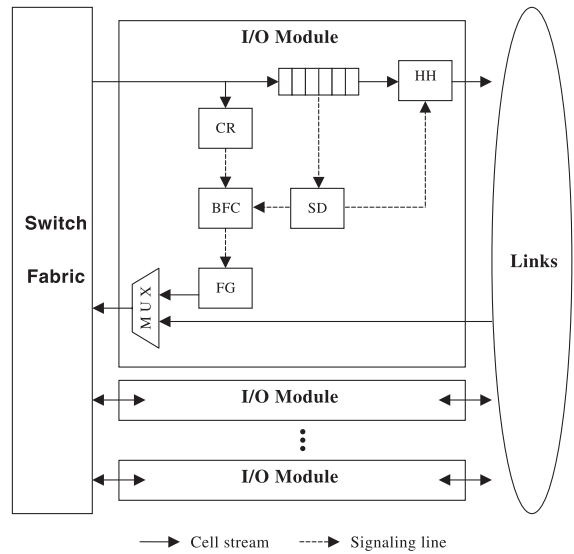


Fig. 6. Functional block diagram of PPFC for switches.

more idle and the system state should be set as non-underloaded. Hence, SES should reject any positive feedback. Conversely, when the switch state change from non-idle to idle, an idle signal carried by RM cell with CI=1 and NI=0 is sent to SES to indicate that the switch is idle. If all switches on one virtual circuit are idle, then the system state should be set as underloaded. Therefore, any positive feedback should be properly processed. The following are the corresponding

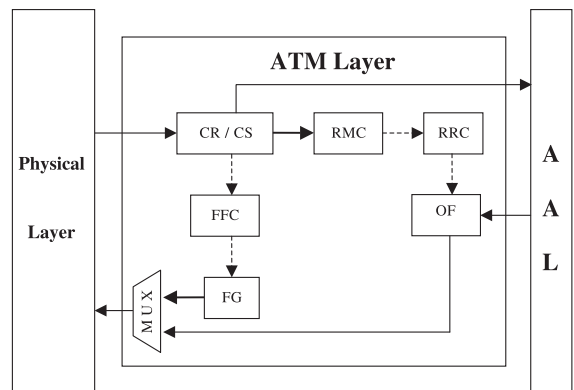


Fig. 7. Functional block diagram of PPFC for end stations.

PPFC algorithms for SES, DES, and the switches. Symbolic constants of PFB\_INTERVAL and NFB\_INTERVAL are two time-constants, which denote the

time interval between each positive feedback and the interval between each negative feedback, respectively.

**Algorithm for PPFC on SES:**

Set System\_State=Underloaded;

Set ACR=MCR;

**while** (there is any data cell has to be transmitted) {

Send data cell with EFCI=0 based on the rate of ACR;

**if** (a RM Cell is received) {

**if** (the RM Cell has CI==0 and NI==1) **then** Set System\_State=Non-Underloaded;

**if** ((the RM Cell has CI==1 and NI==0) and (latest RM cells for other switches all have CI==1 and NI==0)) **then** Set System\_State=Underloaded;

**if** (the RM Cell has CI==1 and NI==1) **then**

$ACR = \max(ACR - (PCR - MCR) \times K\%, MCR);$

**if** ((the RM Cell has CI==0 and NI==0) and (System\_State==Underloaded)) **then**

$ACR = \min(ACR + (PCR - MCR) \times K\%, PCR);$  }

**Algorithm for PPFC on DES:**

**do** { Set Timer=PFB\_INTERVAL;

Set EFCI\_Flag=0;

Countdown Timer;

**while** (Timer>0) {

**if** (data cell with EFCI==1 is received) **then** Set EFCI\_Flag=1;}

**if** (EFCI\_Flag==0) **then** feedback RM cell with CI=NI=0;

} **while** (there is still any data received)

**Algorithm for PPFC on switches:**

Set Switch\_State=Idle;

Set Previous\_Switch\_State=Idle;

Set Queue\_Is\_Growing=False;

```

Set Queue_Is_Shrinking=False;

Set Timer=0;

Set SED_Timer=0;

while (there is still any data cell passing this switch) {

    /* Switch state early detection algorithm. */

    if (SED_Timer==0) then {

         $\Delta Q(i) = Q(i) - Q(i - \text{SED\_TIME\_INTERVAL})$ 

        if ( $\Delta Q(i) > \text{INCREMENT\_THRESHOLD}$ ) then Queue_Is_Growing=True;

        else Queue_Is_Growing=False;

        if ( $\Delta Q(i) < -\text{DECREMENT\_THRESHOLD}$ ) then Queue_Is_Shrinking=True;

        else Queue_Is_Shrinking=False;

        Set SED_Timer=SED_TIME_INTERVAL;}

    Countdown SED_Timer;

    /* Regular switch state detection algorithm. */

    if ((Switch_State==Idle) and ( $Q(i) > Q_{th}$ )) then Set Switch_State=Stable;

    if ((Switch_State!=Idle) and ( $Q(i) < Q_{tl}$ )) then Set Switch_State=Idle;

    if (Switch_State!=Congested) and ( $Q(i) > Q_{hh}$ ) then Set Switch_State=Congested;

    if (Switch_State==Congested) and ( $Q(i) < Q_{hl}$ ) then Set Switch_State=Stable;

    /* Actions done by the switch according to its state. */

    if ((Switch_State==Congested) and (Queue_Is_Shrinking==False)) then {

        if (Timer==0) then {

            Set Timer=NFB_INTERVAL;

            feedback RM cell with CI=NI=1 to SES;}

        Countdown Timer;

    } else Set Timer=0;

```

```

if((Switch_State!=Idle) or (Queueing_Is_Growing==True)) then {
    Mark EFCI=1 on passing data cells;
    if (Previous_Switch_State==Idle) then {
        feedback RM cell with CI=0 and NI=1 to SES;
        Set Previous_Switch_State=Non-Idle;}}
else {
    if (Previous_Switch_State==Non-Idle) then {
        feedback RM cell with CI=1 and NI=0 to SES;
        Set Previous_Switch_State= Idle;}}

```

Most flow control schemes define two states. Therefore, the transmission rates will be frequently adjusted in these two-state algorithms. PPFC defines one additional stable state, which will try to hold the transmission rate fixed. Therefore, the ATM system controlled under PPFC tends to be less wavy, since the transmission rate remains unchanged if this system is stable and is adjusted only if this system is overloaded or underloaded. Additionally, ACR in PPFC is passively controlled by positive and negative feedbacks, which will increase and decrease the transmission rate when the system is under- or overloaded, respectively. This design makes the system under PPFC have a lower ACR and buffer-length oscillation. Even if the system suffers from feedback loss or feedback delay, the ACR remains unchanged. This makes PPFC more defensive against feedback loss or feedback delay even in extremely congested environments. This is because ACR is never automatically increased to make things worse like other flow control schemes using a negative feedback strategy. As to the fairness of PPFC, available bandwidth is fairly assigned to each connection by the following criterion. Let the available bandwidth left for ABR service is  $b_{\text{total}}$  and the minimum required bandwidth for each connection  $i(1 < i < n)$  is  $MCR_i$ . The remaining bandwidth  $b_{\text{left}} = b_{\text{total}} - \sum_{i=1}^n MCR_i$  is then further assigned to each connection by the proportion of their allowed rate range, i.e.  $PCR_i - MCR_i$ . Thus, in PPFC, the real peak cell rate each connection  $i$  can obtain is  $b_i = MCR_i + (PCR_i - MCR_i) \times R$ , where  $R$  is computed based on the above-mentioned rules and is a decimal between 0 and 1.

### 3.2. Implementation

The implementation of the PPFC architecture on ATM switches and end stations is shown in Figs. 6 and 7. In the diagram of intermediate switches, cells come in from a certain link connected to a port handled by the switch fabric. Cells are then forwarded to another port based on the switching table. For the outgoing cells, they are passed to the congestion recognizer (CR) and output buffer simultaneously. The CR checks the EFCI bit on the header of the arriving data cell to recognize whether the congestion notification is tagged or not. This result would then be sent to the backward feedback controller (BFC) and stored. The state detector (SD) with SED support monitors the queue length to make decisions for positive and negative feedback control. If the SD determines that the switching port is non-idle or going to leave the idle state, it will notify the header handler (HH) to mark the EFCI bit on the data cells that are going to leave the port. If the SD determines that the port is congested and not going to relieve, it would not only notify the HH but also notify the BFC. The BFC then periodically makes the feedback generator (FG) to generate a negative feedback to the SESs on all of the input VCs corresponding to the port whose data cells have EFCI=0 until the congestion is relieved. This negative feedback is carried by RM cells with CI=NI=1. They are multiplexed with other cells reaching the port from the physical links. The cells then go into the switch fabric for transmission. Note that the negative feedback is not sent to all VCs



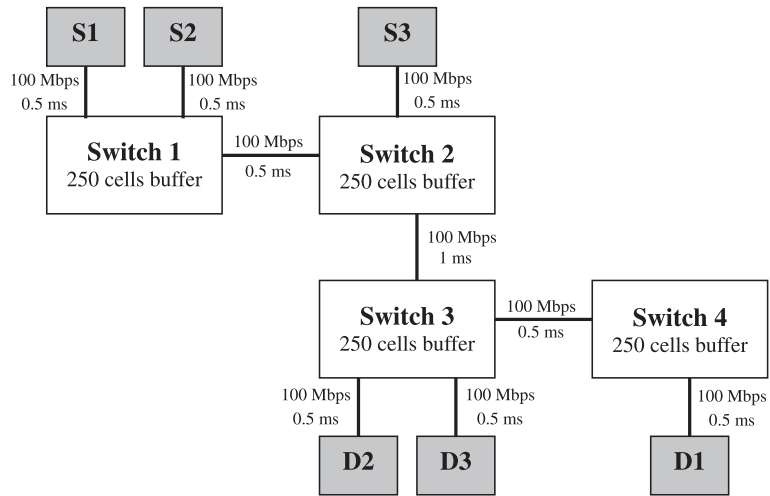


Fig. 8. Network topology for the simulation.

but to selected VCs with unmarked EFCI bits. This would avoid multiple negative feedbacks from different congested nodes to the same SES. To support this feature, an extra bit should be added to the switching table to identify the EFCI status for each VC. The bit is controlled by the BFC, since the EFCI status is directed to the BFC by the CR. The ATM switch should support multicasting, which is out of the scope of this paper.

In the end station diagram, cells entering ATM layer are first handled by the congestion recognizer and cell splitter (CR/CS), which checks whether the

congestion notification is tagged onto the data cell. In the mean time, it also identifies the cell types. If a data cell arrives, CS then transfers the data cell to the ATM adaptation layer (AAL). Furthermore, the EFCI bit of the cell is directed to the forward feedback controller (FFC), which would periodically check the status of the EFCI bits and decide if a positive feedback should be generated by making the FG generate an RM cell with CI=NI=0. If the incoming cell is an RM cell, CS transfers this cell to the RM checker (RMC), which can recognize the CI and NI bits on RM cells and inform the rate regulation controller (RRC) to

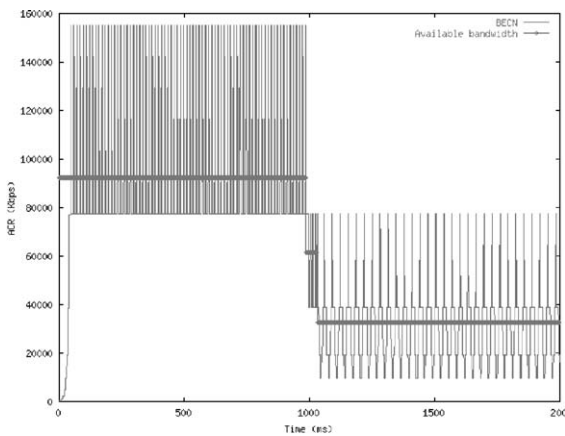


Fig. 9. ACR for BECN.

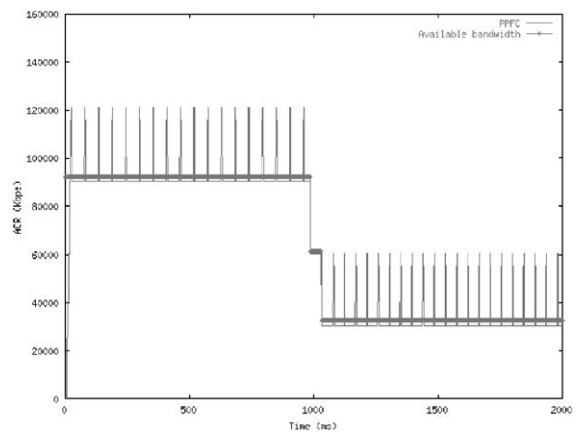


Fig. 10. ACR for PPFC.

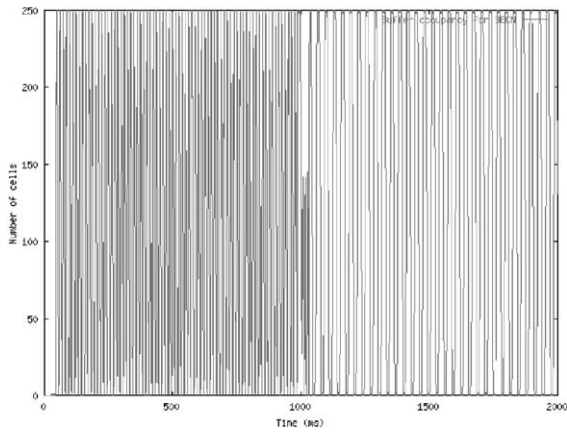


Fig. 11. Buffer occupancy in switch 2 for BECN.

adjust the transmission rate by controlling the output function (OF). Cells from the AAL will be transmitted at the rate controlled by the OF.

#### 4. Simulation results

To evaluate the performance of PPFC, a lot of numerical experiments were made. We considered two different situations: the first involved no feedback loss, whereas the second situation involved a 5% feedback loss. For the purpose of comparison, BECN was chosen, since it has better performance than FECN. The simulation results show that the PPFC algorithm gives better throughput and a lower cell loss rate. Some of these results are described in the following.

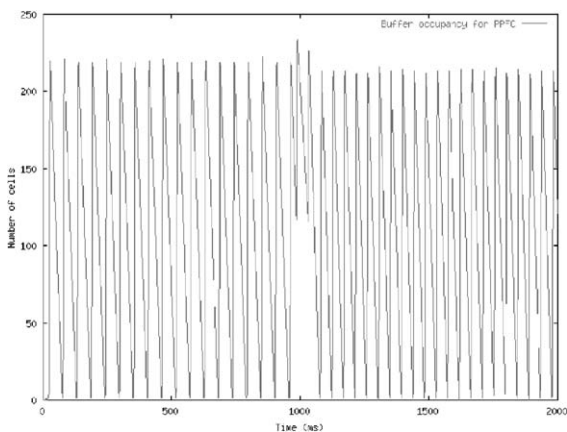


Fig. 12. Buffer occupancy in switch 2 for PPFC.

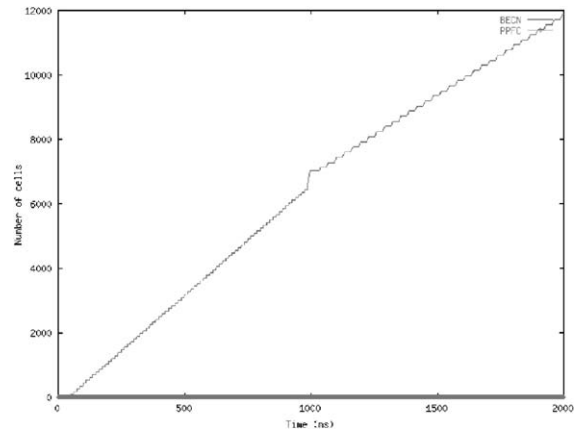


Fig. 13. Number of lost cells in switch 2.

The simulated network topology is given in Fig. 8. There are three ABR traffic source stations (S1, S2, and S3) and the corresponding destination stations (D1, D2, and D3). The MCR and PCR of these sources were set to 1 Mbps and 100 Mbps, respectively. The  $K$  is set to 10%. Each end station was connected using 100 Mbps links to one of four intermediate switches with a buffer size of 250 cells. Although the fixed link bandwidth is adopted in most of the earlier work, we employed variable link bandwidth in this simulation in order to emulate a networking environment in the real world. In this simulation, we assumed that the ABR in the link between switches 2 and 3 would vary randomly from 1 Mbps to 100 Mbps. This is because the remaining bandwidth left by CBR and VBR would never be a

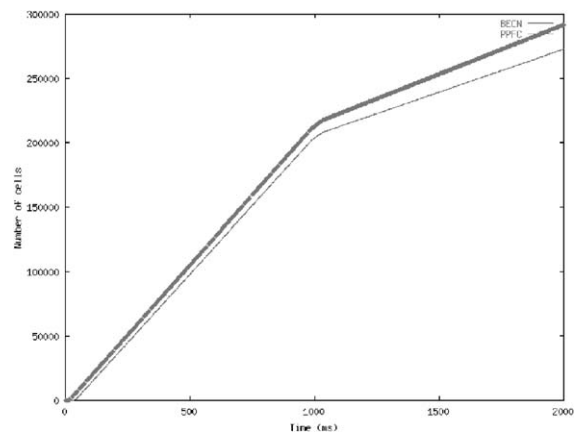


Fig. 14. Total received cells in three DESs.

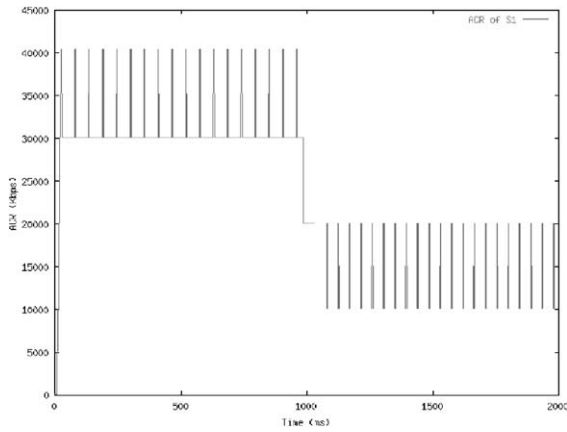


Fig. 15. ACR of S1.

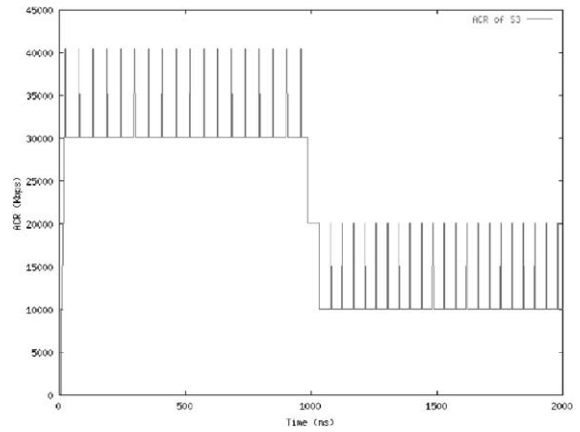


Fig. 17. ACR of S3.

constant in a real system. Furthermore, we chose the propagation delay at 1 ms for the link between switches 2 and 3, and 0.5 ms for all the other links. The single threshold for the queue length on the BECN was set to 100, while  $Q_h$  and  $Q_l$  in the PPFC were set to  $\{110,100\}$  and  $\{60,50\}$ , respectively.

First, we considered the feedback-loss-free environment. The total ACR in the three source end stations is shown in Fig. 9 for the BECN and in Fig. 10 for the PPFC. The ACR of PPFC produced better convergence to the ABR than the BECN did. This also resulted in lower buffer oscillation in the congested switch 2 for PPFC, as shown in Figs. 11 and 12. Most of the time, the queue length for the PPFC switch was kept below 250 cells because of using fast negative feedback to avoid cell loss. ACR

was raised using positive feedback to prevent an empty buffer, which would reduce the utilization. Fig. 13 illustrates the number of lost cells in the congested switch 2. The proposed algorithm has zero cell loss while BECN has 11,880 lost cells in about 2 seconds. The cell loss for the BECN comes from its binary states and exponential rate regulation that cause a larger ACR oscillation during transmission. Fig. 14 illustrates the throughput measured by the total number of received cells in the three destination end stations. As shown, the proposed algorithm has a higher throughput and lower cell loss ratio compared to the BECN in the feedback-loss-free environment. Additionally, to demonstrate fairness of PPFC, individual ACR for three source end stations are shown in Figs. 15–17, respectively.

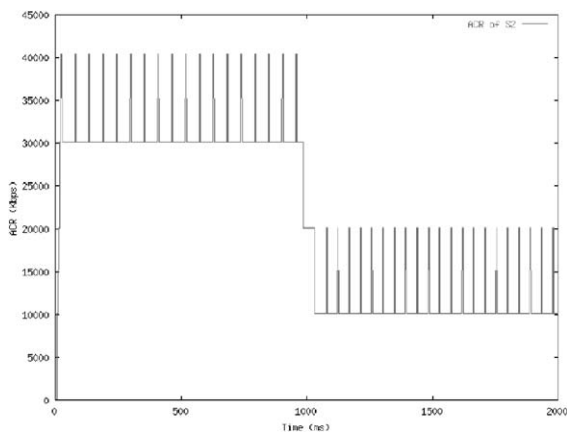


Fig. 16. ACR of S2.

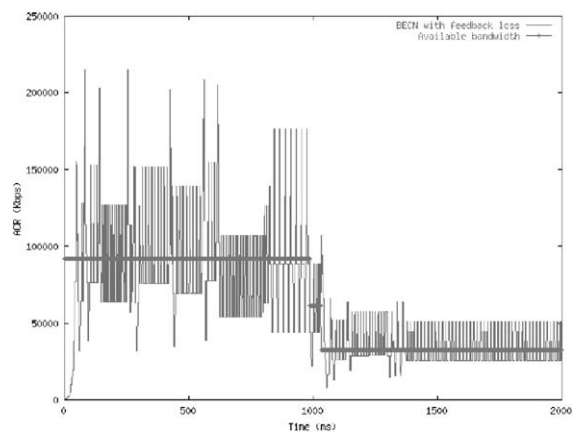


Fig. 18. ACR for BECN with feedback loss.

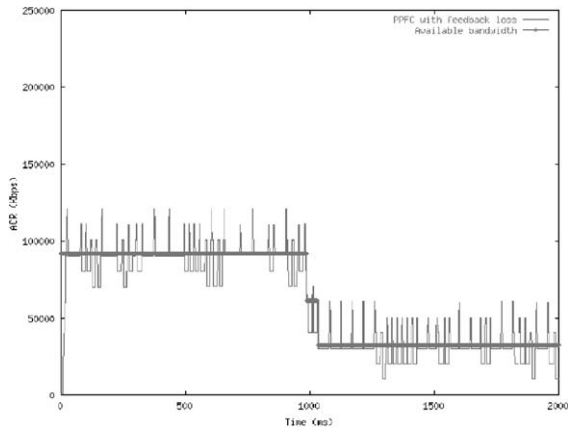


Fig. 19. ACR for PPFC with feedback loss.

Next, we considered the networking environment in which feedback may be lost. The variation of total ACR in three source end stations and buffer occupancy in switch 2 for BECN, and PPFC are shown in Figs. 18–21, respectively. Additionally, Fig. 22 shows the number of cells lost in switch 2 for both algorithms. As can be observed, PPFC lost only 217 cells, while nearly 10,958 cells were lost in the BECN. The throughput for both algorithms is shown in Fig. 23. As shown in the figure, PPFC still has better throughput even though feedback loss occurs. Additionally, in BECN, the DESs totally received 269,297 cells, while exactly 10,958 cells were lost in the congested node for BECN. That is, about 4.07% of the cells were lost. These lost cells would cause some of the received cells to be useless, since they corrupt the correspond-

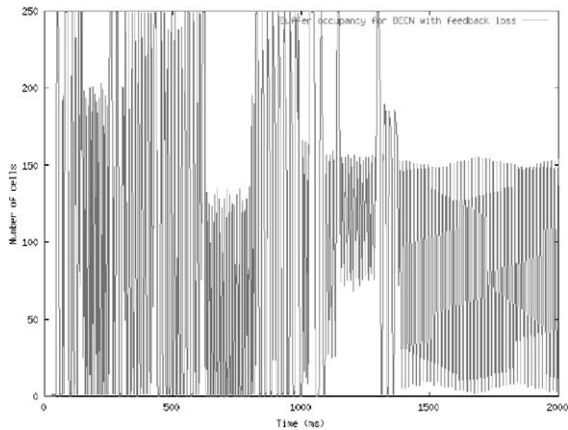


Fig. 20. Buffer occupancy in switch 2 for BECN with feedback loss.

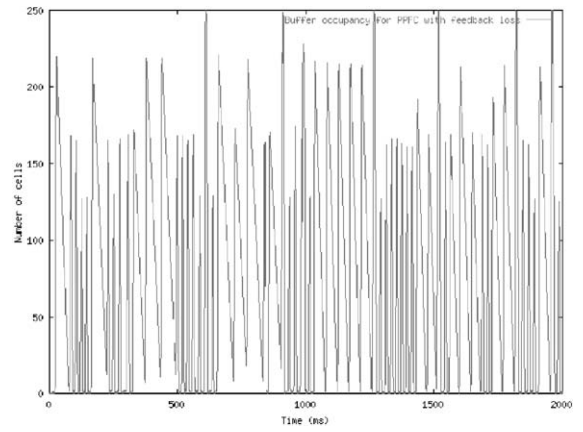


Fig. 21. Buffer occupancy in switch 2 for PPFC with feedback loss.

ing packets of the higher layer. Corrupted packets will be retransmitted by higher layer protocols. This will cause more cells to be retransmitted. Therefore, PPFC also earns better goodput than BECN due to its low cell loss ratio 0.07%.

From the previous results, we know that the performance of PPFC is better than BECN if the single threshold for the queue length on the BECN was set to 100 and  $Q_h$  and  $Q_l$  in the PPFC were set to {110,100} and {60,50}, respectively. However, as described earlier, the values of these thresholds significantly influence the performance of the whole system. Therefore, in Fig. 24, we demonstrate the variation of received cells and lost cells with the choice of threshold for BECN without feedback loss. From the results in Fig. 24, we observe that the

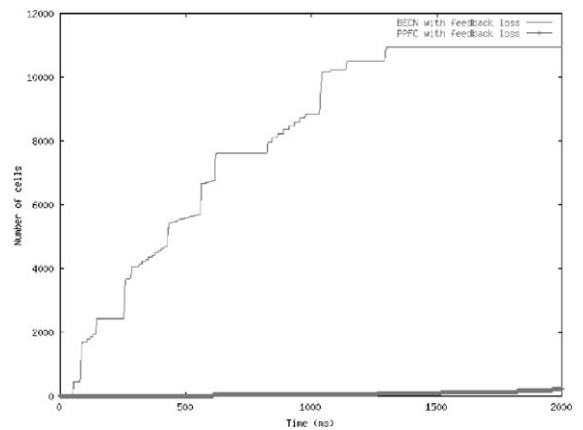


Fig. 22. Number of lost cells in switch 2 with feedback loss.

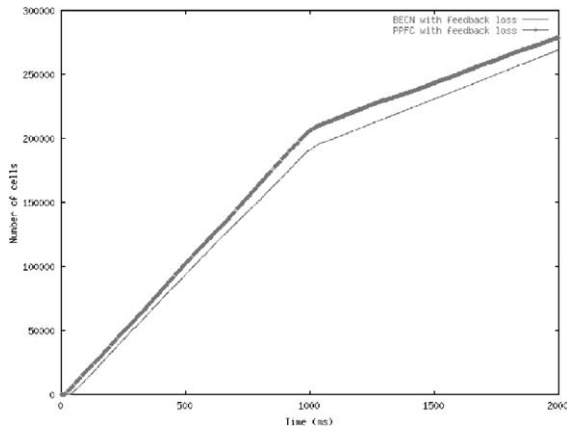


Fig. 23. Total received cells in three DESs with feedback loss.

performance for BECN is very sensitive to the choice of threshold. Although the smaller threshold decreases the number of lost cells, the received cells also decreases. In the experiment, BECN has the lowest loss ratio of 0.000256 with 262,095 cells received on the threshold of 14. As for PPFC, we also demonstrate numerous experimental results of  $Q_l = \{x, x + 10\}$  with  $x = 10 \sim 95$  and  $Q_h = \{y, y + 10\}$  with  $y = 100 \sim 245$ , in Fig. 25. We find that the variation of lost and received cells with both thresholds is very small and most results have very low cell loss ratio and high throughput. In addition, some other choice of  $Q_h$  and  $Q_l$  for PPFC will present better performance than that of  $Q_h = \{110, 100\}$  and  $Q_l = \{60, 50\}$ . Not to mention, the performance of  $Q_h = \{110, 100\}$  and  $Q_l = \{60, 50\}$  for

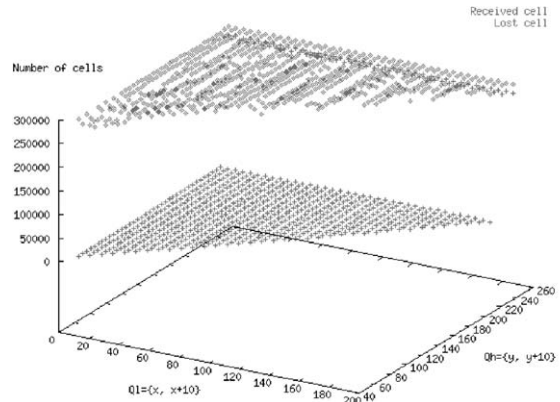


Fig. 25. Variation of lost/received cells with  $Q_l$  and  $Q_h$  for PPFC.

PPFC is better than that of any choice of threshold for BECN.

### 5. Conclusions and future work

In this paper, we proposed an improved rate-based flow control algorithm, which controls the transmission rate effectively at the source end stations for the ABR traffic on ATM networks. This algorithm employs positive and negative feedback to control the transmission rate. The transmission rate is increased, decreased, or remains unchanged while the system is underloaded, overloaded, or stable. Additionally, we added state early detection to predict the upcoming status of the switch. PPFC can therefore control the transmission rate by making a look-ahead observation. This algorithm also avoids the over-regulation problem by using the idea of threshold bands. We also proposed a design of architecture for PPFC implemented on both ATM switches and end stations. As shown, PPFC can easily be implemented at low cost. Most importantly, the simulation results show that this algorithm offers better throughput and a lower cell loss ratio compared to BECN. Buffer oscillation is relieved due to the controlled ACR in our approach. Furthermore, PPFC provides excellent defense against feedback loss, while the BECN performance is significantly degraded by the lost cells. We think that reducing ramp-up and cut-down time may enhance the rate regulation framework in the

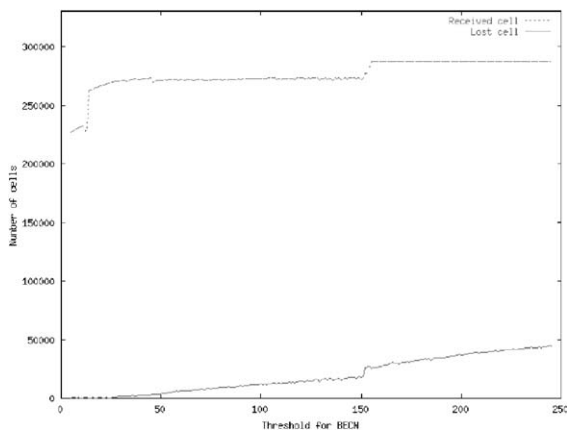


Fig. 24. Variation of lost/received cells with threshold for BECN.

proposed algorithm. In that case, throughput should be increased even more.

## References

- [1] F. Bonomi, K. Fendick, The rate-based flow control framework for the available bit rate ATM service, *IEEE Netw.*, (1995) 25–39.
- [2] H.T. Kung, R. Morris, Credit-based flow control for ATM networks, *IEEE Netw.* (1995) 40–48.
- [3] W.K. Lai, D.R. Shiu, M.C. Liou, J.Y. Tsai, Fair and reliable hop-by-hop flow control, *Comput. Commun.* (1999) 1227–1233.
- [4] C.F. Su, G.D. Veciana, J. Walrand, Explicit rate flow control for ABR services in ATM networks, *IEEE/ACM Trans. Netw.* (2000) 350–361.
- [5] S. Kamolphiwong, A.E. Karbowiak, H. Mehrpour, Survey flow control in ATM networks: a survey, *Comput. Commun.* (1998) 951–968.
- [6] B.A. Makrucki, On the performance of submitting excess traffic to ATM networks, *Proc. IEEE GLOBECOM* (1991) 281–288.
- [7] B.A. Makrucki, Explicit forward congestion notification in ATM networks, *Proc. TriComm '92* (1992).
- [8] P. Newman, Backward explicit congestion notification for ATM local area networks, *Proc. IEEE GLOBECOM* (1993) 719–723.
- [9] P. Newman, Traffic management for ATM local area networks, *IEEE Commun. Mag.* (1994) 44–50.
- [10] L. Roberts, et al., Closed-loop rate-based traffic management, *ATM Forum/94-0438R1* (1994).
- [11] L. Roberts, Enhanced PRCA (proportional rate-control algorithm), *ATM Forum Contribution 94-0735R1*, 1994.
- [12] K.Y. Siu, H.Y. Tzeng, Adaptive proportional rate control (APRCA) with intelligent congestion indication, *ATM Forum Contribution 94-0888* (1994).
- [13] Y.C. Chen, C.T. Chan, S.C. Hu, On the effective traffic control of ABR services in ATM networks, *IEICE Trans. Commun.* (1998) 417–430.
- [14] W.K. Lai, J.Y. Tsai, A flow control scheme on ATM networks with max-min fairness, *Comput. Commun.* (1999) 543–555.
- [15] Y.C. Lai, Y.D. Lin, Choice of high and low thresholds in the rate-based flow control scheme, *IEE Proc. Commun.* (1999) 95–101.
- [16] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Trans. Netw.* (1993) 397–413.



**Cheng-Yuan Ku** was born in Taipei, Taiwan on October 14, 1965. He received the BS degree in Control Engineering from National Chiao Tung University, Taiwan in 1987 and the MS and PhD degrees in Electrical Engineering from Northwestern University in 1993 and 1995, respectively. From 1995 to 1999, he was with Department of Information Engineering, I-Shou University. Since 1999, he joined the Department of Information Management, National Chung Cheng University. His current research interests include computer network, cryptography, and network management. He is the member of INFORMS, IEEE, and IEICE.



**Lain-Chyr Hwang** was born in Taiwan on 26 December 1965. He received his BS degree in Electrical Engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan in 1988, his MS degree in Communication Engineering in 1990, and his PhD in Electronic Engineering in 1994 from National Chiao Tung University, Hsinchu, Taiwan. From 1995 to 1996, he was with Telecommunications Laboratories, Chunghwa Telecom, Taiwan, as an associate researcher. There, he was involved in network planning of the intelligence network and personal communication services. He was an associate professor in the Department of Management Information Systems of the I-Shou University, Taiwan, from August 1996 through July 2000, and in the Department of Electrical Engineering, since August 2000. His research interests include performance evaluation and QoS control of multimedia communication networks.



**Hsin-Tzung Wu** received the BE and the ME degrees in Institute of Information Engineering from I-Shou University, Kaohsiung, Taiwan in 1997 and 2000, respectively. He is currently working toward the PhD degree in the Institute of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan. His current research interests include Internet, high-speed networks, congestion control in ATM networks, and shape coding in object-based image processing.