ELSEVIER

Computer Communications 24 (2001) 1272–1282

computer communications

www.elsevier.com/locate/comcom

A simplified approach based on source control for ATM ABR service

Chun-Liang Lee*, Yaw-Chung Chen, Jin-Ru Chen

Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC

Received 18 February 2000; revised 4 October 2000; accepted 10 October 2000

Abstract

The available bit rate (ABR) service is designed to provide efficient support of data traffic in ATM networks. A variety of rate-based flow control approaches have been proposed to support ABR service. Most of these approaches improve both the fairness among active connections and the link utilization by putting more and more complexity into a switch. In this article, we propose a flow control approach in which part of rate-calculation work is moved from switches to end-systems. As a result, the proposed approach reduces the rate-calculation effort in the switch. This may reduce the switch complexity and thus the implementation cost. Furthermore, it mitigates the difficulty for setting the measurement interval in switches, as well as features lower and more stable queue occupancy in the network. Simulation results are presented to demonstrate that the proposed approach achieves better performance than ERICA + , a well-known switch algorithm.

Keywords: Asynchronous transfer mode; Available bit rate service; Traffic management; Rate-based flow control

1. Introduction

Asynchronous transfer mode (ATM) networks support five service categories: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non-real-time VBR (nrt-VBR), available bit rate (ABR), and unspecified bit rate (UBR) [1]. The former three provide guaranteed quality of services, while the latter two offer best effort services. The ABR service intends to support bursty data applications. A source end-system in ABR service is allowed to send data at a rate no less than its minimum cell rate (MCR), which could be specified in the connection setup phase. The source adjusts its transmission rate according to the feedback information from the network. The feedback information is carried in resource management (RM) cells, which are periodically sent by the source and returned by the destination. An RM cell on its way from the source to the destination is called a forward RM (FRM) cell. Once this RM cell is returned by the destination, it is called a backward RM (BRM) cell. As stated in Ref. [1], the source should send an RM cell once *Nrm*-1 consecutive data cells have been sent. The default value of the parameter *Nrm* is 32.

Since the transmission rate of a source is determined based on the feedback information, it is crucial for the network to provide appropriate feedback to sources. Several switch algorithms have been proposed in Refs. [2,7,8,10–14]. The objectives of these algorithms are to quickly acquire the available bandwidth, and to achieve a fair allocation of the available bandwidth among active ABR connections. An observation of the evolution of ABR flow control shows that more and more complexity has been put on switches for achieving better performance. This increases the implementation cost of the switch and thus prevents the ABR service from being widely deployed by service providers. In addition, with the rapid advance in fiber technology, it is difficult for a complicated switch to provide wire-speed-switching capability if complicated services were deployed.

In this paper, we propose a rate-based flow control approach in which a part of the rate calculation work is moved from switches to end-systems. By performing queue occupancy control in source end-systems, our approach helps reduce the switch complexity and achieve lower and more stable queue occupancy in the network. The rest of this paper is organized as follows. Section 2 reviews the previous work on rate-based flow control. Then our proposed flow control approach is addressed in Section 3. Simulation environment and numerical results are presented in Section 4. Finally, Section 5 concludes the work.

2. Previous work on rate-based flow control

In the early development stage of ABR service, the

---

* Corresponding author. Tel.: +886-3-573-1851; fax: +886-3-572-7842.
E-mail address: leecl@csie.nctu.edu.tw (C.-L. Lee).

0140-3664/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(00)00344-3

network congestion indication was conveyed by the explicit forward congestion indication (EFCI) bit in the cell header. The destination of a connection sends RM cells back to the source at appropriate time, and this gives the source a permission to increase its transmission rate [2,10,12]. The appropriate timing for sending the RM cells back to the source varies among different approaches. The common advantage of these approaches is the simplicity. However, this class of approaches suffers from the 'beat down problem', in which a connection passing through more congested nodes gets lesser opportunity to increase its rate than those passing through fewer congested nodes. Therefore, it is difficult to achieve good fairness for all connections. A possible way to alleviate this problem is supporting per-VC queueing in switches, but it is too complicated to be cost effective.

Another class of approach, the explicit rate (ER) scheme, provides precise control to achieve better performance. Instead of using one bit to indicate the network status, the switch marks the actual rate it can support in RM cells. Several ER schemes have been discussed extensively in the ATM Forum. For example, enhanced proportional rate control algorithm (EPRCA) [13] computes a mean allowed-cell-rate (MACR) for all virtual connections (VCs) using a running exponential weighted average. When a switch encounters the congestion, it sets the CI bit and modifies the ER field of the BRM cell in the VC whose current cell rate (CCR) is larger than MACR. EPRCA features better fairness and link utilization than the previous approaches. Its weakness is the rate fluctuation caused by the rate calculation, which uses the CCR field in every RM cell received to derive the weighted average value. Since the calculated MACR may be far apart from the actual allowed cell rate, it is still not very precise. Furthermore, it may cause unnecessary rate oscillation.

The explicit rate indication for congestion avoidance (ERICA) [11] was presented at the ATM Forum in 1995. ERICA requires a switch to measure the load factor of each link at every fixed time interval, which is called *switch measurement interval*. The load factor, z, is derived through the following equation:

$$z = \frac{\text{Input rate}}{\text{ABR capacity}}.$$

The input rate is the incoming ABR traffic measured in a measurement interval, and the capacity of ABR service is the remaining link capacity not used by the higher priority service classes, such as CBR and VBR traffic. In order to fully utilize the link bandwidth quickly, the variable $VCShare_i$, which denotes the calculation result for connection i, is computed as follows:

$$VCShare_i = \frac{\mathrm{CCR}_i}{z}.$$

Obviously, if each source adjusts the transmission rate to its corresponding *VCShare* value, the load factor will be equal to one. However, the calculation of *VCShare* does not take the fairness problem into account because a source with a larger CCR will get a higher *VCShare* than those with smaller CCRs. The fairness among all active VCs can be achieved by distributing the ABR capacity over them fairly. The fair share of each VC is computed as

$$FareShare = \frac{\text{ABR capacity}}{\text{Number of active ABR connections}}.$$

The value of *FairShare* denotes the minimum rate that a VC can transmit. By the combination of *VCShare* and *FairShare*, a switch modifies the ER fields in a BRM cell as follows:

$$\text{ER calculated} = \max(VCShare, FairShare),$$

$$\text{ER} = \min(\text{ER calculated, ER in the BRM cell}).$$

Thus, ERICA guarantees the fairness among active VCs and utilizes the network efficiently. However, ERICA does not consider the queue occupancy in the intermediate switches, and this may cause the queues to grow unbounded. In order to overcome this drawback, the target utilization in ERICA must be set to a value smaller than one (e.g. 0.9 or 0.95) to drain the queues. However, this degrades its efficiency because the link utilization is restricted by the target utilization parameter.

ERICA + [11], an improved version of ERICA, was proposed to overcome such weakness. ERICA + takes the queue occupancy as a secondary metric. Based on the observation that non-empty queues imply 100% link utilization, the goal of ERICA + is to keep the queuing delay of a switch at a pre-defined threshold ($T0$). Depending on the available capacity for ABR service, the value of $T0$ would be translated into a queue length ($Q0$), which is shown as follows:

$$Q0 = \text{ABR capacity} \times T0.$$

Then the switch adjusts the available capacity, which is used for rate calculation, according to the relationship between the current queue length (q) and $Q0$. If $q > Q0$, the available capacity will be decreased to drain the queue. Otherwise, it will be increased for allowing the queue length to grow up to $T0$. Another two parameters (a and b) are used to control how aggressively the switch increases and decreases the available capacity, respectively. The parameter a determines the amount of the available capacity reduction when q is larger then $Q0$. The larger the value of a is, the larger the amount of capacity reduction will be. The parameter b determines how much excess capacity would be allocated when q is smaller than $Q0$. A larger b enables the switch to increase the queue length faster. Although larger values of a and b make ERICA + more sensitive to the queuing delay, it may cause more severe rate oscillation. In order to limit the amount of capacity reduction, the queue drain limit factor (QDLF) is defined, and it denotes the maximum percentage of the available capacity which can be used to drain the queue. Therefore, the value of QDLF is between 0 and 1.

3. Proposed end-to-end rate control approach

3.1. Motivation

Although ERICA + can fully utilize links, it requires a switch to measure the load factor and the number of active VCs over a measurement interval as ERICA does. In addition, the switches need to perform more calculation for queue occupancy control. This makes switches more complicated. Moreover, it is difficult to determine the number of active VCs for allocating the bandwidth fairly. Although a method was proposed in Ref. [9] to deal with this problem, it may further complicate the switch design.

Furthermore, it is hard to choose a proper value as the measurement interval. A shorter interval provides faster response to the current status of the switch, but it may consume too much processing power of the switch if the switch has to perform lots of computation work at the end of each measurement interval. In addition, it may fail to estimate the number of active VCs, and thus the fairness cannot be guaranteed. On the other hand, a larger measurement interval reduces the overhead of rate calculation and can estimate the number of active VCs more precisely. However, it leads to a slow response to the traffic changes and thus may cause severe performance degradation during congestion.

In order to reduce the switch complexity as well as to ease the setting of the measurement interval, our algorithm moves the queue occupancy control from switches to source end-systems. The main purpose of queue occupancy control is to prevent the severe performance degradation caused by queue overflow. Because once a cell is dropped by a switch, it is necessary for the source to retransmit the entire higher layer protocol data unit (PDU) which contains this dropped cell. Therefore, a dropped cell will usually cause the source to retransmit many cells that have already been successfully received by the destination. Furthermore, queue occupancy control affects the queuing delay of a cell along its end-to-end path. It has to keep the queue occupancy in a switch always larger than zero because a non-empty queue implies 100% link utilization. On the other hand, it has to keep the queue occupancy below a certain value in order to achieve its main purpose, say, avoiding the packet dropping.

Queue occupancy control also guarantees the fairness among active connections. In a queuing node with common FIFO service discipline, the bandwidth sharing of a connection depends on its ratio of the queue occupancy. If every connection keeps a nearly equal queue occupancy in the queuing node, a fair bandwidth sharing can be achieved.

3.2. Queue occupancy estimation

Our proposed rate control approach performs the queue occupancy control in sources. However, a source cannot directly obtain the queue occupancy information. In our previous work [5,6], the concept of virtual queue occupancy (VQO) was proposed. The objective of VQO is to enable a *receiver* to estimate the queue occupancy in the network and provide feedback to the source for determining the appropriate transmission rate. In this paper, we propose a new mechanism to derive the queue occupancy. The new mechanism is different from the old one in two aspects, which are discussed as follows:

1. *Deployment* — The queue occupancy estimation is performed in *sources* rather than in receivers. Thus it is easier to deploy the new approach because the cooperation between the source and the receiver is not essential.
2. *Implementation complexity* — The new mechanism does not require that a source sends cells in a specific pattern. This simplifies the source function and makes the proposed algorithm more feasible to implement.

We now present the mechanism of queue occupancy estimation. Recall that an ABR source sends out an RM cell once a certain amount of data cells have been sent out. These RM cells will be returned by the receiver and eventually sent back to the source; therefore, they can be used to derive the network status. In our approach, a source estimates the queue occupancy in the network by monitoring RM cells. In order to reduce the control overhead, a source keeps only one *monitor RM cell* in the network. In fact, a monitor RM cell is just a regular RM cell whose sequence number and departure time are recorded for rate calculation purpose. In other words, an outgoing RM cell is chosen as a monitor RM cell if there is no monitor RM cell in the network. Once a monitor RM cell is sent, the source will record the departure time and start counting the number of cells sent before it receives the returned RM cell. When the source receives a monitor RM cell, it calculates the round-trip time of this cell by subtracting the cell's departure time from the current time. The measured round-trip time consists of two parts: the *fixed delay* and the *queuing delay*. The former comprises the round-trip propagation delay, the transmission delay and the protocol processing delay. This part is treated as a fixed value, and considered as the minimum of the round-trip times ever measured. The latter is the cell queuing delay in intermediate nodes along the round-trip path. Obviously, the queuing delay is the difference between the measured round-trip time and the fixed delay.

By considering the whole network as a queuing node, the queue occupancy (N) for VC_i can be estimated as follows [3]:

$$N = \mu T, \tag{1}$$

where μ denotes the service rate of VC_i, and T denotes the required queuing delay for a newly arrived cell. Since T is already known to the source, the queue occupancy can be

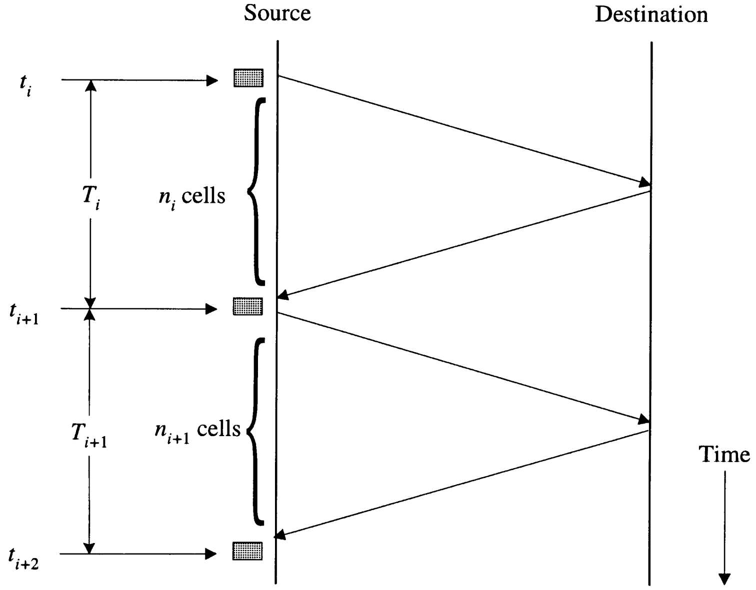

Fig. 1. Service rate estimation.

estimated through Eq. (1) if μ is also available. The way to obtain μ is as follows.

Fig. 1 shows the scenario of service rate estimation. The variable t_i represents the time at which the ith monitor RM cell is sent out, and the variable T_i denotes the time interval between the departure times of the ith and the $(i+1)$th monitor RM cells. Let n_i be the number of cells sent during T_i. When the source receives the $(i+1)$th monitor RM cell, the round-trip time T_{i+1} can be derived. By considering the network as a single service node, the arrival of the $(i+1)$th monitor RM cell at the source denotes that the cells sent during the time period T_i have been served by the network. As a result, by recording the number of cells sent during T_i, the service rate μ can be derived through the following equation:

$$\mu = \frac{n_i}{T_{i+1}}. \tag{2}$$

To explain the meaning of Eq. (2), suppose that the sending rate of the source remains the same during time periods T_i and T_{i+1}, and the network service rate becomes smaller than the sending rate of the source since $T = t_i$. As a result, T_{i+1} will be larger than T_i, which causes the network service rate calculated by Eq. (2) at $T = t_{i+2}$ to be smaller than that at $T = t_{i+1}$. A similar method was also used in Ref. [4] to implement TCP Vegas for Linux. Once a source receives a monitor RM cell, it first derives the service rate using Eq. (2) and then uses Eq. (1) to estimate the queue occupancy.

Since the queue occupancy estimation is triggered by a monitor RM cell, a source has to consider the situation where a monitor cell gets lost. Because each RM cell has its unique sequence number and the source knows the sequence number of the current monitor RM cell, when the source receives an RM cell whose sequence number is larger than the recorded sequence number of the monitor RM cell, it can infer that the monitor RM cell was lost. A possible way is to use the received RM cell to estimate the queue occupancy. The drawback of this approach is that the queue occupancy estimated will be larger than the actual value since the measured round-trip time is larger than the actual round-trip time. To eliminate the estimation error, a source can keep the departure time of all outstanding RM cells to measure a correct round-trip time. Obviously, this requires the source to keep more information.

3.3. Queue occupancy control

In our proposed approach, a source tries to keep the queue occupancy in the network at a pre-defined level. This ensures not only the full utilization of the link but also aids in avoiding the queue overflow, which may lead to severe performance degradation. Once a source receives a monitor RM cell, it can estimate the queue occupancy in the network using the scheme discussed in Section 3.2. Also, the source calculates the round-trip time (*rtt*) of this RM cell and the network service rate (μ). Suppose that the estimated queue occupancy is *vqo*, and the pre-defined target queue occupancy is α. If the source wants to keep the queue occupancy at α, the new allowed cell rate (ACR) can be calculated as

$$\mathrm{ACR} = \mu + \frac{\alpha - vqo}{rtt}. \tag{3}$$

If the estimated queue occupancy is smaller than the control target (i.e. $vqo < \alpha$), this represents that the source can increase its transmission rate. As shown in the above equation, the new ACR calculated will be a value larger than the network service rate μ. If the transmission rates of other sources are kept the same, the new ACR will cause the queue occupancy to reduce from vqo to α after one round-trip time. The case $vqo \geq \alpha$ is similar to that discussed above.

3.4. Switch behavior

Since the queue occupancy control is performed in the source, the switch behavior is very simple in our approach. A switch measures the load factor z on each link periodically. Once the switch receives a BRM cell of VC_i, it first calculates the ER as

$$\text{ER calculated} = \frac{\mathrm{CCR}_i}{z}.$$

This is like the calculation of *VCShare* in ERICA. The value of ER cannot be larger than the capacity for ABR service. Thus, we have

$$\text{ER calculated} = \min(\text{ER calculated, ABR capacity}).$$

Finally, in order to provide the rate information that the bottleneck switch can support, the ER field of the RM cell can be set to

$$\text{ER} = \min(\text{ER calculated, ER of the RM cell}).$$

3.5. Source behavior

When an RM cell is to be transmitted, the source sets this RM cell as a monitor RM cell if there is no monitor RM cell in the network. When a source receives a monitor RM cell, it calculates the rtt of this RM cell, the network service rate (μ) and the vqo. If vqo is smaller than the pre-defined threshold α, it indicates that the source can increase its ACR. To determine the new ACR, the source has to calculate two rates (ACR_1 and ACR_2). ACR_1 is calculated by applying the standard rule of source [1], which is as follows:

$$ACR_1 = \min(\text{ACR} + RIF \times PCR, \text{ ER in the RM cell}).$$

ACR_2 is calculated for queue occupancy control, which is discussed in Section 3.3. In order to avoid the unnecessary rate oscillation, we introduce a new parameter, $factor_1$, whose value is set between 0 and 1. As a result, ACR_2 is calculated through the following equation:

$$ACR_2 = \mu + factor_1 \times \frac{\alpha - vqo}{rtt}.$$

The value of $factor_1$ determines the response time for a source to adjust the queue occupancy, and further details regarding how to set the parameter is discussed in Section 3.6. The source then sets its ACR as the maximum of these two rates:

$$\text{ACR} = \max(ACR_1, ACR_2). \tag{4}$$

In the case where the network is under light load, ACR_1 tends to be larger than ACR_2. The purpose of ACR_1 is to fully utilize the network if each source sets its ACR to the corresponding ACR_1. If the network is either in congestion or almost fully utilized, ACR_1 tends to be smaller than ACR_2. ACR_2 is used by a source to control its queue occupancy in the network. Setting the ACR to ACR_2 allows a source to increase its rate for achieving the control target of queue occupancy. Thus, even if the ER carried in the arriving RM cell is smaller than the new transmission rate, the source is still able to get its fair rate sharing. As a result, our approach provides efficient bandwidth usage via ACR_1 as well as controls the queue occupancy via ACR_2.

If vqo is larger than α, the source adjusts its transmission rate as follows:

$$\text{ACR} = \min\left(\mu - factor_2 \times \frac{vqo - \alpha}{rtt}, \textit{ ER in the RM cell}\right). \tag{5}$$

Similar to the case of $vqo \leq \alpha$, Eq. (5) adjusts the ACR by considering both the ER in the RM cell and the rate calculated for queue occupancy control, while $factor_2$ is a parameter whose value is between 0 and 1.

3.6. Selection of the parameters

In our proposed approach, the source has three parameters: $factor1$, $factor2$ and α. This section discusses how to choose the parameter values.

The parameters $factor_1$ and $factor_2$ determine how quickly a source adjusts its rate to reach the control target of queue occupancy. The parameter $factor_1$ is used to increase the queue occupancy while $factor_2$ is used for the opposite purpose. A large value of $factor_1$ (or $factor_2$) enables the source to increase (or decrease) its queue occupancy quickly; however, this may cause more serious rate oscillation. On the contrary, a small value of $factor_1$ (or $factor_2$) alleviates the rate oscillation but needs more time for the source to reach its target queue occupancy. How to choose the values for $factor_1$ and $factor_2$ is a tradeoff between the stability and the response time. Generally, the parameter $factor_1$ is smaller than $factor_2$ since it is more urgent to reduce the queue occupancy than to avoid the over-reduction of rate. However, it is preferable to increase the queue occupancy conservatively for avoiding queue overflow which is usually caused by excessively increasing the queue occupancy.

The parameter α is the control target of the queue occupancy. It is similar to the parameter $T0$ in ERICA + . A larger α can make our approach more efficient when the rate oscillation is serious. This is because a large α is more likely to prevent the queue from dropping off to empty. Obviously, it is limited by the buffer size of the switch.

3.7. Comparison between ERICA + and our approach

Our proposed approach differs from ERICA + in the following aspects:

1. Switch complexity — At the end of each measurement

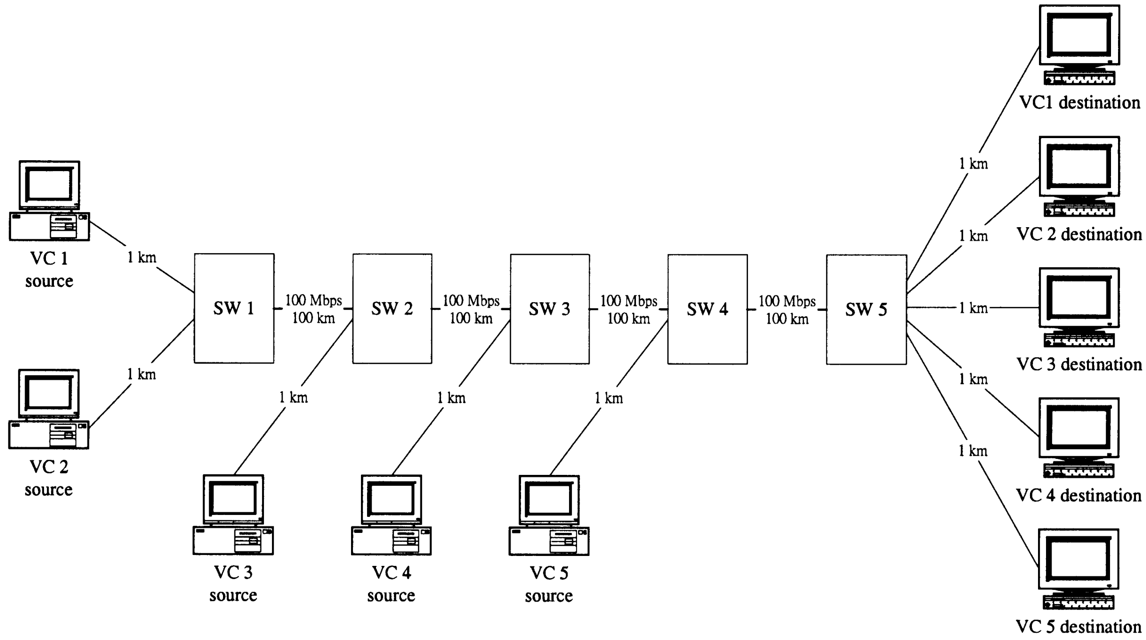

Fig. 2. Simulation configuration.

interval, ERICA + needs to measure the load factor of each link, to estimate the number of active connections and to perform calculations for queue control. To estimate the number of active connections, a switch must check whether to update the variables used for counting every time a cell is received. In contrast, our approach only needs to measure the load factor. Obviously, with our approach a switch does much less work for rate calculation than ERICA + .

2. The setting of measurement interval — As described in Section 3.1, it is hard for ERICA + to choose a proper measurement interval due to the tradeoff between a fast response to the network status and a precise estimation of the number of active connections. Since our approach does not need to estimate the number of active connections, it is simpler than ERICA + in selecting a proper measurement interval.
3. Queue occupancy control — ERICA + controls the queue occupancy in switches, and the performance of queue control is determined by the setting of measurement interval. However, the most appropriate measurement interval for a switch depends on the round-trip times of those connections passing through it. In contrast, our approach controls the queue occupancy in sources based on the round-trip time. As a result, our approach is less sensitive to the setting of measurement interval. For example, suppose that a switch chooses a long measurement interval, which may be caused by a parameter-setting error or the system limitation. For the connection whose round-trip time is shorter than the measurement interval, ERICA + fails to provide a fast enough feedback to the source, and thus it may lead to poor performance. Our algorithm, however, does not have this kind of problem because the source will adjust its rate according to the estimated queue occupancy.

Table 1
Source parameters

VC	MCR (Mbps)	PCR (Mbps)	ICR (Mbps)
VC 1–VC 4	0	100	1
VC 5	0	4	1

4. Performance evaluation

4.1. Simulation environment

The network model used throughout this work is a parking lot configuration, which is commonly used in literature to illustrate the fairness problem [8]. A configuration with five switches is shown in Fig. 2.

In our simulation, each link has a capacity of 100 Mbps and the distance between adjacent switches is 100 km. The length for all network-access links is set to 1 km. Five ABR VCs destined to SW5 are set up. VC1 and VC2 originate from SW1 and traverse all five switches, while VC3, VC4 and VC5 start at SW2, SW3 and SW4, respectively. When all the five VCs are active, the bottleneck in this configuration is the link between SW4 and SW5. Thus, all five VCs traverse through the bottleneck link. The longest end-to-end round trip propagation time is 4.02 ms in this configuration.

Table 1 lists the parameter values of a source. The source

Table 2
Simulation parameters of ERICA +

Parameter	Value
δ	0.1
$T0$	3.5 ms
A	1.15
B	1.05
QDLF	0.5
Measurement interval	1 ms

of VC5 is a constrained source whose peak cell rate (PCR) is only 4 Mbps while the PCR of other sources is up to the link capacity (i.e. 100 Mbps). The initial cell rate (ICR) of each source is set to 1 Mbps to demonstrate the process of rate contention in different approaches. Table 2 lists the parameter settings in an ERICA + switch. The parameter values are chosen according to the suggestions in Ref. [11]. Our approach needs other parameters for queue control. The target queue occupancy α is set to 50 cells, and the values of $factor_1$ and $factor_2$ are set to 0.25 and 0.5, respectively.

4.2. Numerical results

The performance of our proposed approach is evaluated based on the ACR of each source and the queue length in each switch. As regards the queue length of each switch, we only show the queue lengths of SW3 and SW4 because the queue lengths of both SW1 and SW2 are very small and even zero most of the time during the simulation.

Figs. 3 and 4 show the ACR of each source for ERICA + and the proposed approach, respectively. As observed from Fig. 3, the ACR of VC5 reaches a stable rate quickly because its PCR is only 4 Mbps, which is smaller than the fair share rate of the bottleneck link (i.e. 20 Mbps). The ACRs for VC1–VC4 fluctuate between 19 and 30 Mbps, which are around the fair share rate (i.e. $(100 - 4)/4 = 24$). The reason for the ACR fluctuations is that ERICA + reduces the ER of each VC dramatically once the queue length in a switch is larger than the pre-defined threshold. However, to quickly drain the queue, it may over-reduce the ER and thus cause unnecessary rate oscillation. As compared with ERICA + , in our approach the ACR of each source is more stable. This is because our approach controls the queue occupancy more precisely. Since ERICA + spends certain effort to measure the number of active connections, the converging time for the ACR to reach the fair share rate is shorter than ours. However, our approach features very small rate fluctuations. This means that our algorithm performs flow control efficiently, and it is much simpler than ERICA + .

Figs. 5 and 6 show the queue lengths under different approaches. Since the ACRs in ERICA + fluctuate dramatically, the queue lengths in SW3 and SW4 are not stable. The queue length in the bottleneck switch varies between 0 cells and 1400 cells. The main reason for this severe fluctuation of queue length is that ERICA + controls the queue occupancy at every measurement interval, and it does not

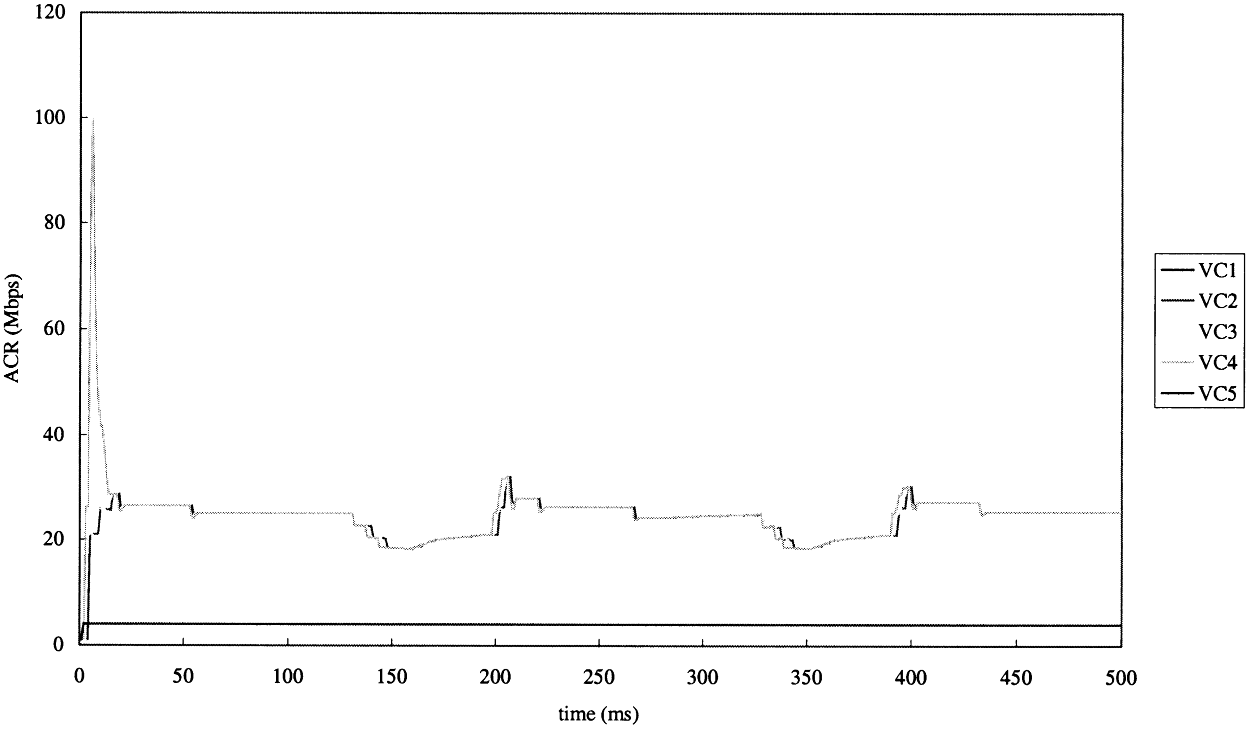

Fig. 3. Allowed cell rates of ERICA + ($T0 = 3.5$ ms).

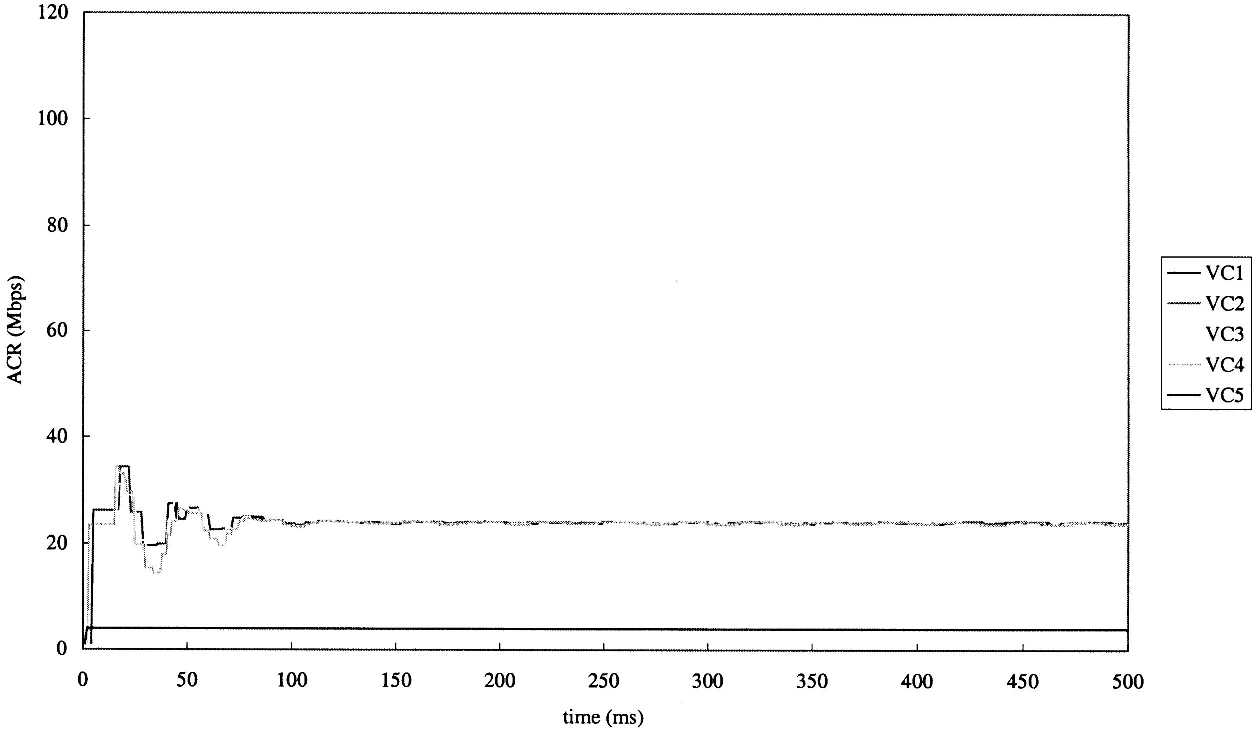

Fig. 4. Allowed cell rates of our proposed algorithm.

consider the difference in round-trip times for different sources. When a switch detects that the queue length is larger than the pre-defined threshold, it immediately reduces the transmission rates of all connections passing through it. Because of the delayed effect of the control process, it may over-reduce the queue occupancy before it detects that the queue length is smaller than the threshold. This can be observed in Fig. 5, where the

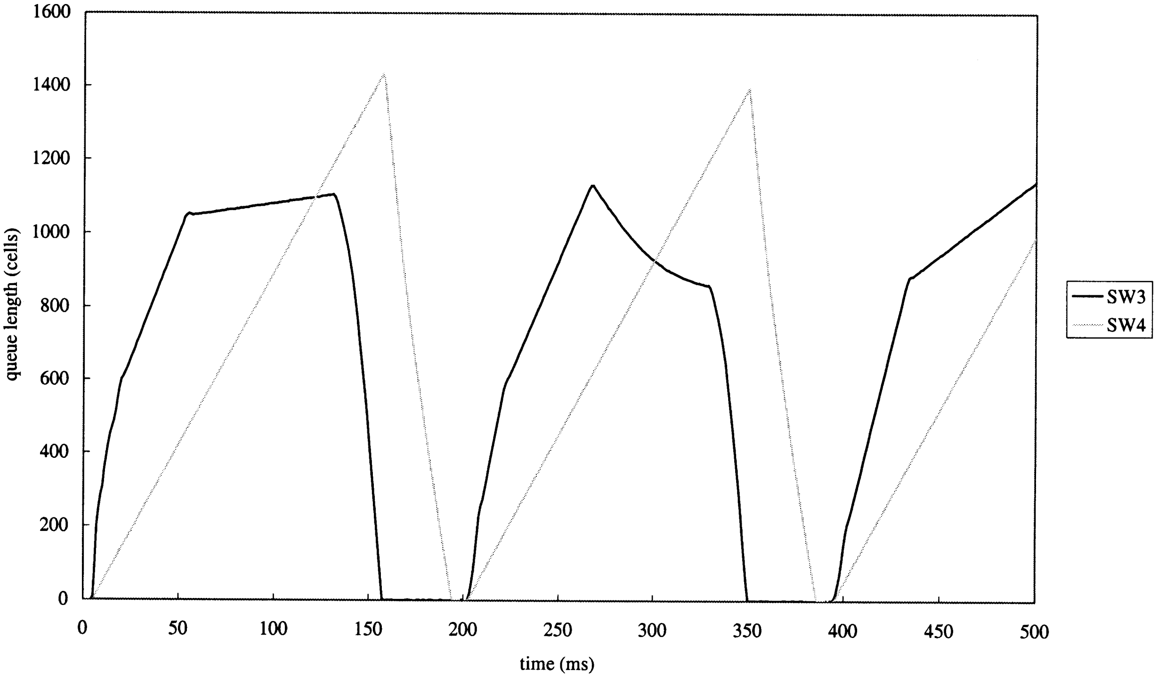

Fig. 5. Queue lengths of ERICA + $(T0 = 3.5\ \mathrm{ms})$.

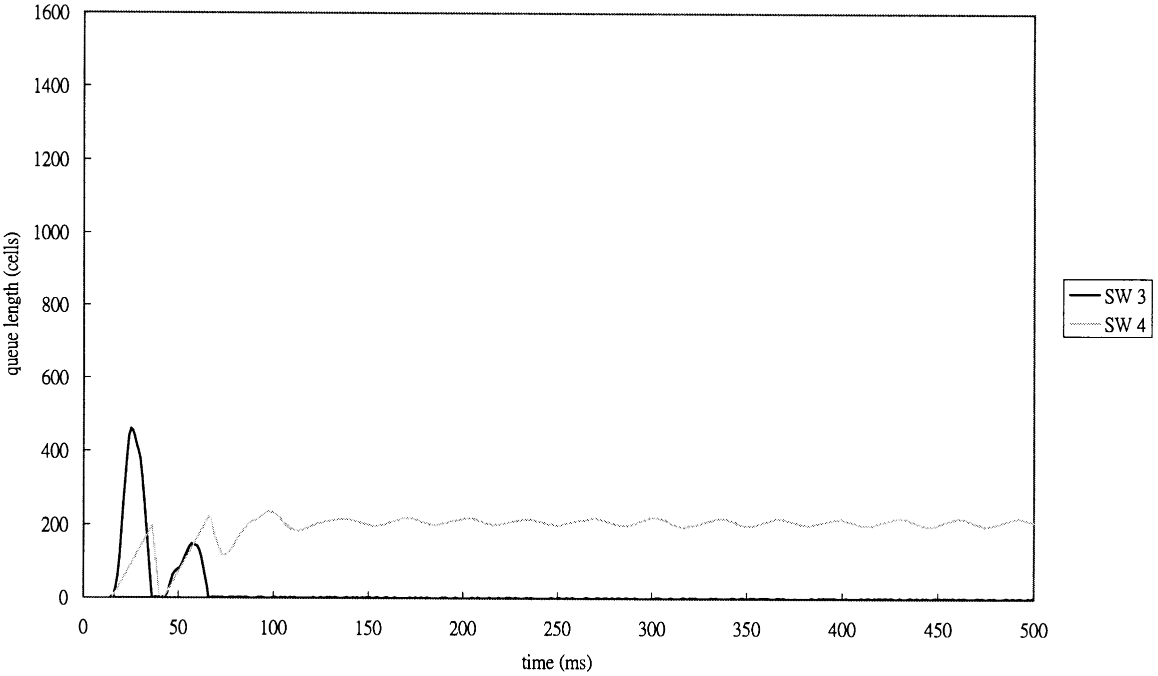

Fig. 6. Queue lengths of our proposed approach.

queue length of SW4 is reduced to 0 at both $T = 200$ ms and $T = 390$ ms. In contrast, our approach performs the queue control in sources, and each source adjusts its rate to control the queue occupancy on per-round-trip time basis. Therefore, the queue length of the bottleneck switch is much smaller and more stable than that in ERICA + . The queue length in SW3 is near zero in the steady state. This shows that our algorithm

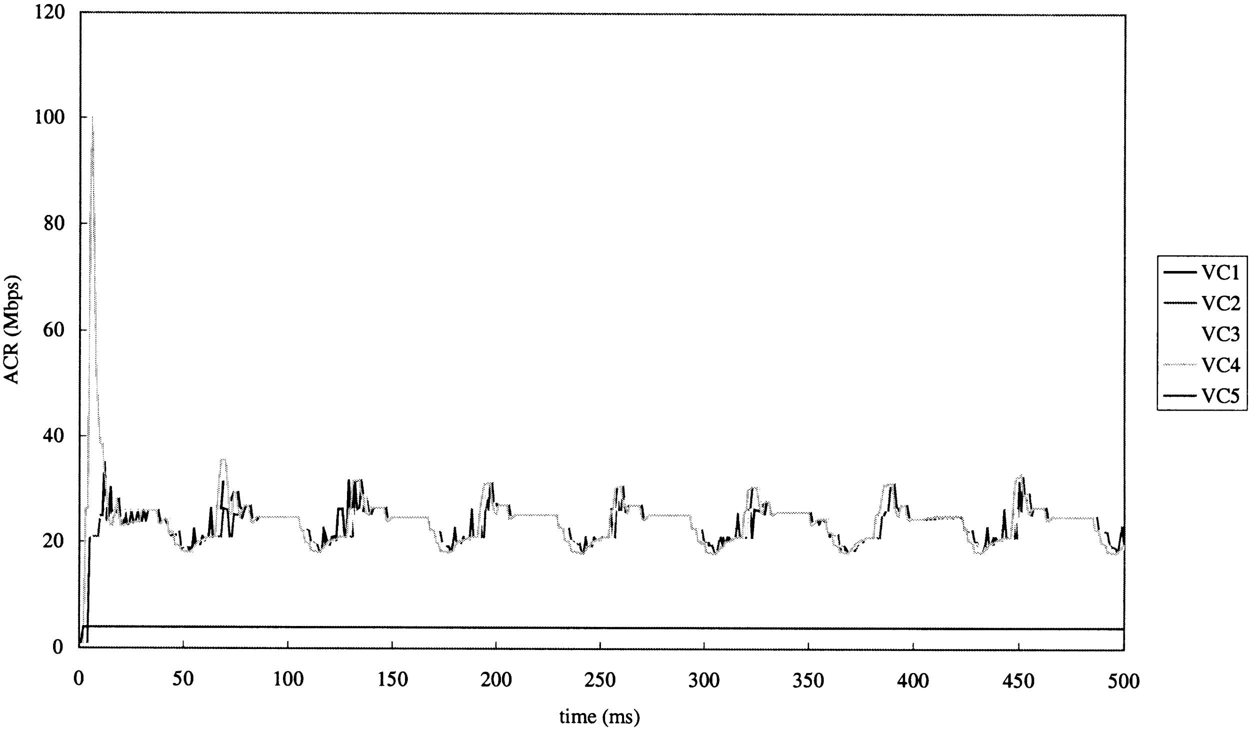

Fig. 7. Allowed cell rates of ERICA + ($T0 = 1$ ms).

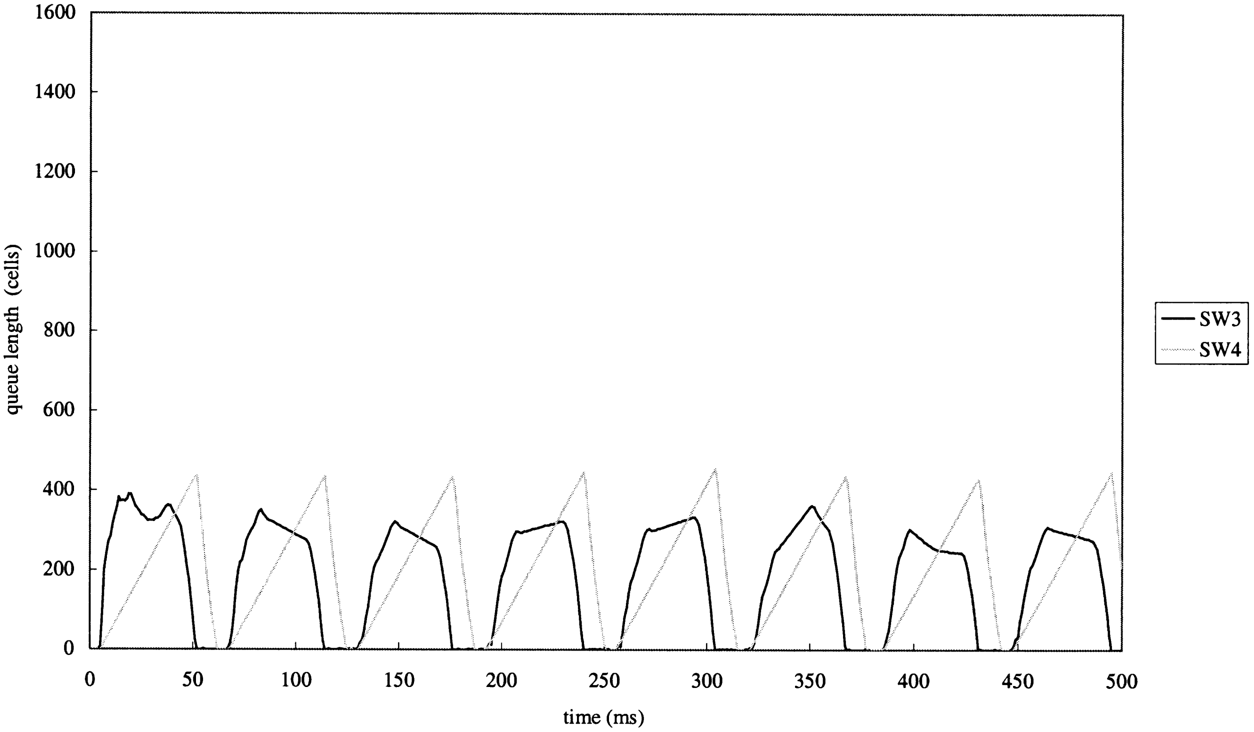

Fig. 8. Queue lengths of ERICA + ($T0 = 1$ ms).

features smaller queuing delay and delay variation while still keeping 100% bottleneck link utilization.

As stated in Ref. [11], the parameter $T0$ introduced in ERICA + is used to replace the target utilization parameter of ERICA. It determines the target queuing delay of VCs that pass through a switch. Therefore, reducing the value of $T0$ may reduce the queue length of the switch as well. However, it may also cause not only severe rate oscillation in sources but also under-utilization of the bottleneck link. In order to observe the effect, we set the value of $T0$ to 1 ms and keep other parameters unchanged. The simulation results are shown in Figs. 7 and 8. As we expect, the ACRs of sources shown in Fig. 7 oscillate more severely than that shown in Fig. 3. The maximum queue length of bottleneck switch is reduced from 1400 cells to 450 cells, but the queue length drops to 0 cells more often. Thus, it leads to worse utilization of the bottleneck link. This experiment shows that it is difficult to choose a proper value of $T0$ in ERICA + with the tradeoff between the queuing delay and the link utilization.

5. Conclusions

In this paper, we propose a flow control approach for ABR services. Different from most existing approaches, our approach performs queue occupancy control in end-systems rather than in switches. With our approach, the switches only need to measure the load factor of each link. It is unnecessary for a switch to determine the number of active VCs for distributing the available bandwidth fairly. As a result, the complexity and implementation cost of switches can be greatly reduced. Reducing the complexity of switches means cutting down the implementation cost. This benefits the design of a switch as well as enhances its performance. Moreover, it overcomes the difficulty in selecting the measurement time interval and avoids the tradeoff between the fast response to load changing and the precise calculation for the number of active VCs. As compared with ERICA +, it also simplifies the work a switch has to perform either when it receives an RM cell, or when the timer of the measurement interval expires. With precise control of the queue occupancy in the network, our approach is able to avoid unnecessary rate fluctuation. It also features smaller and more stable queue length in the bottleneck switch. Thus, our approach achieves smaller delay variation, which is crucial for supporting multimedia applications over the ATM ABR services [15].

Acknowledgements

The authors would like to thank the anonymous reviewers who provided helpful feedback on the manuscript.

References

[1] The ATM Forum, The ATM Forum traffic management specification

version 4.0, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[2] A.W. Barnhart, Baseline performance using PRCA rate-control, ATM Forum Contribution 94-0597, July 1994.

[3] L.S. Brakmo, L.L. Peterson, TCP Vegas: end to end congestion avoidance on a global Internet, IEEE J. Select. Areas Commun. 13 (10) (1995) 1465–1480.

[4] N. Cardwell, B. Bak, A TCP Vegas implementation for Linux, http://www.cs.washington.edu/homes/cardwell/linux-vegas/.

[5] J.R. Chen, Y.C. Chen, C.T. Chan, A distributed end-to-end rate control approach for ABR services, Proc. IEEE GLOBECOM 4 (11) (1998) 2446–2451.

[6] J.R. Chen, Y.C. Chen, Delay and analysis of queuing delay control for end-to-end rate control algorithm, IEICE Trans. Commun. 10 (10) (1999) 1577–1585.

[7] Y.C. Chen, C.T. Chan, S.C. Hu, On the effective traffic control of ABR services in ATM networks, IEICE Trans. Commun. 2 (2) (1998) 417–430.

[8] F.M. Chiussi, Y.T. Wang, An ABR rate-based congestion control algorithm for ATM switches and per-VC queuing, Proc. IEEE GLOBECOM 2 (11) (1997) 771–778.

[9] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, B. Vandalore, On determining the fair bandwidth share for ABR connections in ATM networks, Proc. IEEE Int. Conf. Commun. (1998) 1485–1491.

[10] M. Hluchyj, N. Yin, On closed-loop rate control for ATM networks, Proc. IEEE INFOCOM (1994) 99–108.

[11] R. Jain, S. Fahmy, S. Kalyanaraman, R. Goyal, ERICA switch algorithm: a complete description, ATM Forum Contribution 96-1172, August 1996.

[12] H. Hsiaw, K. Fendick, F. Bonomi, L. Roberts, A. Jain, B. Makrucki, T. Tofigh, A.W.Barnhart, L. Gun, B. Holden, G. Fedorkow, J.N. Daigle, D. Kline, H. Suzuki, G. Ramamurthy, P. Newman, N. Giroux, R. Kositpaiboon, D. Hughes, M.G. Hluchyj, G. Garg, K.Y. Siu, H.Y. Tzeng, N. Yin. Closed-loop rate-based traffic management, ATM Forum Contribution 94-0438R1, July 1994.

[13] L. Roberts, Enhanced PRCA (proportional rate-control algorithm), ATM Forum Contribution 94-0735R1, August 1994.

[14] K.Y. Siu, H.Y. Tzeng, Adaptive proportional rate control (APRCA) with intelligent congestion indication, ATM Forum Contribution 94-0888, September 1994.

[15] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, M. Goyal, QoS and multipoint support for multimedia applications over ATM ABR service, IEEE Commun. Mag. (1) (1999) 53–57.