Journal of Southeast University (English Edition) Mar. 2002 Vol.18 No.1 ISSN 1003—7985

# A Fuzzy Flow Control Approach for ABR Service in ATM Networks

Zhang Xiaolin* Zhang Sabing Wu Jieyi

(CIMS Center, Southeast University, Nanjing 210096, China)

**Abstract:** The explicit rate flow control mechanisms for ABR service are used to share the available bandwidth of a bottleneck link fairly and reasonably among many competitive users and to maintain the buffer queue length of a bottleneck switch connected to the link at a desired level in order to avoid and control congestion in ATM networks. However, designing effective flow control mechanisms for the service is known to be difficult because of the variety of dynamic parameters involved such as available link bandwidth, burst of the traffic, the distances between ABR sources and switches. In this paper, we present a fuzzy explicit rate flow control mechanism for ABR service. The mechanism has a simple structure and is robust in the sense that the mechanism's stability is not sensitive to the change in the number of active virtual connections (VCs). Many simulations show that this mechanism can not only effectively avoid network congestion, but also ensure fair share of the bandwidth for all active VCs regardless of the number of hops they traverse. Additionally, it has the advantages of fast convergence, low oscillation, and high link bandwidth utilization.

**Key words:** ATM network, congestion prevent, flow control, fuzzy logic

A major development in high-speed networking is the emergence of broadband integrated service digital networks (B-ISDN) and asynchronous transfer mode (ATM). Due to the unpredictable fluctuations and burst of traffic within ATM networks, congestion can occur frequently. Hence, it is necessary to design appropriate congestion control mechanisms to ensure the promised quality-of-service (QoS) is met.

Several service classes have been defined in ATM for the support of traffic with different QoS requirements. The available bit rate (ABR) service class was defined for the support of best-effort applications. This service class does not provide any strict guarantees, it attempts to minimize the cell loss at the expense of delay and allows applications to fully utilize the available bandwidth in the network by adjusting their instantaneous transmission rates to the available capacity.

The ATM Forum Traffic Management Committee has completed the specification of a rate-based flow control framework to satisfy this objective[1]. Thus, the rate-based flow control mechanisms for ABR service are designed not only to avoid and control network congestion but also to guarantee the available bandwidth being shared fairly among all ABR competitive users in ATM networks[1,2].

The various rate-based flow control mechanisms can be classified broadly two categories depending on the feedback mechanism employed: binary feedback mechanism[3] and explicit rate (ER) mechanism. In the elapsed few years, a number of ER mechanisms have been proposed[4] due to their reported advantages over binary feedback mechanism. In Ref. [5], the authors suggest that formal methods should be applied to the design of closed loop flow control in ATM networks. There are two kinds of formal methods which is usually used to design the flow control mechanisms in the literatures. One is based on traditional control theory[6], and the other is based on intelligence technique[7].

The fuzzy logic is a formal computational intelligence technique and has been used to efficiently solve several ATM problems such as usage parameter control[8], connection admission control[9]. Therefore, as part of our ongoing research project focusing on the development of intelligent control techniques for ATM networks, in this paper, we present an ER flow control mechanism based on fuzzy logic control technique for avoiding and controlling congestion in ATM networks by keeping the ABR queue length around a desired value. Our primary objectives in designing such a flow control mechanism are to achieve: rapid convergence and low oscillation of the controlled network parameters such as the buffer queue length of the switch, fair share of the available bandwidth for all active virtual connections (VCs) regardless of the number of hops

Received 2001-12-17.

* Born in 1963, male, doctor.

they traverse, high utilization of the link bandwidth, and robusticity in the sense that the mechanism's stability is not sensitive to the change of the network parameter such as the number of active VCs.

The rest of this paper is organized as follows. In section 1, we outline briefly the operational principles of our fuzzy mechanism. In section 2, we describe the framework of a fuzzy flow controller (FFC) used in the mechanism, guidelines of design for the FFC, and the fairness processing. In section 3, by using the several simulation results, we discuss and evaluate the performance of our proposed fuzzy mechanism. Finally, we present our conclusion in section 4.

# 1 Fuzzy Flow Control Mechanism Model

In this paper, the fuzzy mechanism mainly consists of ABR source end system (SES), ABR destination end system (DES), the FFC and fairness processing module (FPM) operating in the switch. The SES and the DES behaviors are both compliant with the specification[1]. Fig.1 shows the model of the mechanism. In Fig.1, ACR denotes allowed cell rate, $C$ is the bandwidth of the link, QT is the desired queue length threshold, $e(n)$ denotes the error between QT and the instantaneous queue length $q(n)$ which is measured and fed to FFC at the end of the $n$th sampling period, $\Delta e(n)$ is the change of $e(n)$.

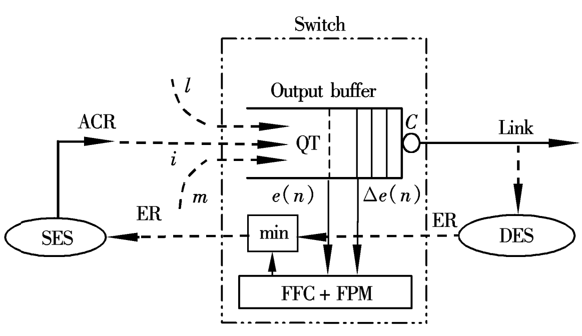



**Fig.**1 Fuzzy flow control mechanism model for ABR

The fuzzy mechanism operates as follows: the SES periodically sends a "forward" resource management (FRM) cell to the DES every NRM data cells. This FRM cell contains several fields, mainly including the current cell rate (CCR) field and the ER field that carries a 15-bit floating point number representing the explicit rate. Before transmitting each FRM cell, the SES initializes the ER field to its peak cell rate (PCR) negotiated at connection set up, and the CCR field to its current ACR. Upon receipt of the FRM cell, the DES returns it to the source and this FRM becomes a "Backward" RM (BRM) cell. According to the included FFC and FPM, switch computes an average fair share of the available capacity for per active VC at the end of each sampling period. When a BRM cell arrives at the switch, ER value encapsulated in this BRM cell is compared with the current fair share value, and then the result which is a new explicit rate for the BRM cell's corresponding VC is fed back to the SES of the VC in the ER field of the BRM. The SES adjusts its ACR to the feedback information ER encapsulated in its current accepted BRM cell, but the ACR must range between minimum cell rate (MCR) which is a rate negotiated at connection set up and PCR.

# 2 FFC and FPM Designs

In this section, we briefly introduce fuzzy logic and fuzzy logic controller (FLC), and show how they can be used to represent an unknown mathematical model. We will then discuss the design of our FFC and FPM.

## 2.1 Overview of FLC

Fuzzy logic is a method for representing information in a way that resembles natural human communication, and for handling this information in a way that is similar to human reasoning. The fuzzy sets theory describes incomplete or vague concepts which might be difficult to formulate mathematically. FLCs, like expert systems, are used to model human experience and human decision making behavior. In a FLC the input-output relationship is expressed by using a set of linguistic rules or relational expressions. As shown in Fig.2, a FLC basically consists of four important parts including a fuzzifier, a defuzzifier, an inference engine and a rule base.

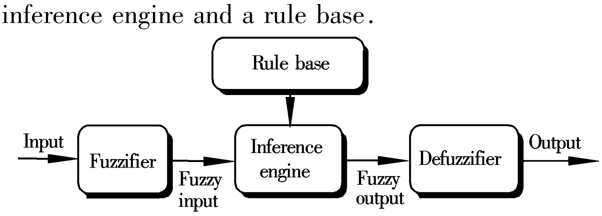



**Fig.**2 Block diagram of a FLC

The role of the fuzzifier is to map each crisp input to fuzzy sets defined by their membership functions depending on the degree of "possibility" of the input. The goal of the defuzzifier is to map the output fuzzy sets to a crisp output. It combines the different fuzzy sets with different degrees of possibility to produce a single output. The fuzzy inference engine defines how the FLC should infer through the rules in the rule base to determine the output fuzzy sets. The rule base which consists of a set of if-then rules is the heart of the FLC.

In a double-input-single-output (DISO) FLC, the form of an if-then rule is as follows:

Rule: if $X_1$ is $A$ and $X_2$ is $B$ then $Y$ is $C$

where $X_1$, $X_2$ and $Y$ are linguistic variables, and $A$, $B$, $C$ are their linguistic values respectively. Each linguistic value is a fuzzy set and is characterized by its membership function. The rule defines a relation between the linguistic values $A$, $B$ and $C$. In other words, it performs a mapping from fuzzy input state-space to a fuzzy output value.

## 2.2 FFC design

FFC is a fuzzy logic controller. Like other FLCs, FFC design involves definition of the input variables which can indicate the buffer current state and its future behavior, definition of the output variables which can be used to control the ABR traffic, selection of shapes of membership functions for linguistic values of these variables, rule base design, selection of defuzzification method and the inference engine.

### 2.2.1 Definition of input and output variables

In this paper, FFC, based a DISO FLC model and the linguistic information stored in the rule base, uses two input variables $e(n)$ and $\Delta e(n)/T$ to calculate an output variable $\Delta \mathrm{er}(n+1)$ at the end of the $n$th sampling period. Among these variables, $e(n)$ is used to capture the current state of the buffer, and its change rate $\Delta e(n)/T$ is used to predict the future behavior of the buffer. Here, $T$ is the sampling period. $\Delta \mathrm{er}(n+1)$ is an explicit rate increment in next sampling period, it will be added to the available capacity $r(n)$, which is computed at the end of previous period, and then $r(n+1)$ is obtained, which is the available capacity in next sampling period. By using FFC, we try to control the error signal $e(n)$ and its change $\Delta e(n)$ nears zero, and so the instantaneous queue length $q$ will converge at the desired value QT.

### 2.2.2 Rule base design

For the sake of having a set of minimum number of linguistic rules with sufficient accuracy to achieve an acceptable performance, the term sets of input linguistic variable $E$ and EC corresponding respectively to $e(n)$ and $\Delta e(n)/T$ are both defined as {Negative Big(NB), Negative Small(NS), Zero(ZE), Positive Small (PS), Positive Big(PB)}, and the term set of output linguistic variable ΔER corresponding to $\Delta \mathrm{er}(n+1)$ is the same as above term sets. Since triangular and trapezoidal shaped membership functions offer more computational simplicity, we select them for the fuzzy sets shown in Fig.3 – Fig.5, where $B$ is the size of the buffer, $Q_E = M_E \times \mathrm{QT}$, $Q_{EC} = M_{EC} \times \mathrm{QT}/T$, $Q_\Delta = M_\Delta \times C$. In the next simulation, we choose $M_E = M_{EC} = 0.08$, $M_\Delta = 0.5$, QT = 150 cells, $B$ = 5 000 cells, $T$ = 1 ms.

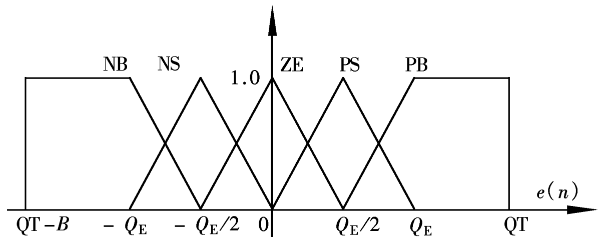



**Fig.**3 Linguistic values and their membership functions of $E$

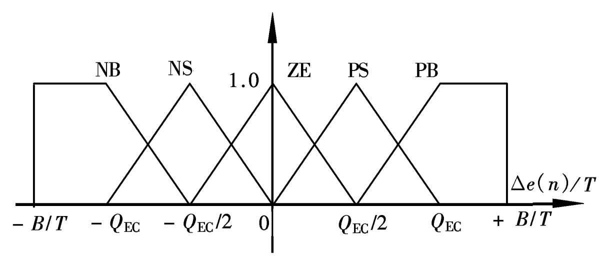



**Fig.**4 Linguistic values and their membership functions of EC

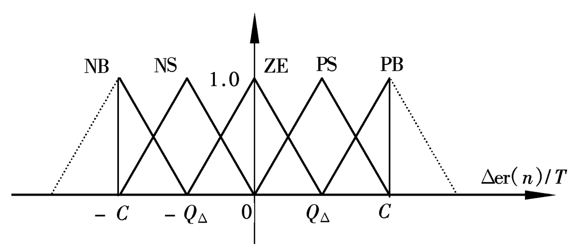



**Fig.**5 Linguistic values and their membership functions of ΔER

The design of linguistic rules is based on our experience and beliefs on how the mechanism should behave, and these rules are fine tuned by observing the progress of simulation. Tab.1 shows these rules, and the form of a rule marked with arrow is as follows:

Rule: if $E$ is negative small and EC is positive small then ΔER is zero

**Tab.**1 FFC rules

| ΔER | | EC | | | ↓ | |
|---|---|---|---|---|---|---|
| | | NB | NS | ZE | **PS** | PB |
| | NB | NB | NB | NS | NS | ZE |
| → | **NS** | NB | NS | NS | **ZE** | PS |
| | ZE | NS | ZE | ZE | ZE | PS |
| $E$ | PS | NS | ZE | PS | PS | PB |
| | PB | ZE | PS | PS | PB | PB |

### 2.2.3 Selection of the defuzzification method

Defuzzification is related to the interference engine. In this paper, we select the min-max interference engine due to its simplicity and practicability. Many defuzzification methods are available such as the gravity method, the maximum membership method and the weighted mean method. To construct our FFC, the gravity method is chosen. Therefore, the crisp output $\Delta er(n+1)$ of FFC is obtained as

$$\Delta er(n+1)=\frac{\sum_{j=1}^{N} w_j \times \mu_j(w_j)}{\sum_{j=1}^{N} \mu_j(w_j)} \tag{1}$$

where $N$ is equal to 5 which is the number of output fuzzy set; $\mu_j(w_j)$ is the final solved membership value for the $j$th output fuzzy set; $w_j$ is the center value of the output fuzzy set.

### 2.3 FPM design

In this subsection, we describe fairness processing. From Eq.(1), the available capacity $r(n+1)$ can be obtained.

$$r(n+1)=r(n)+K\Delta er(n+1) \tag{2}$$

where $K$ is a scale factor which is 0.08 in next simulations. In order to avoid big overshoot which is yielded in initialization time, the nonlinear maximum function is introduced to impose bound on the $r(n+1)$. Thus, we obtain

$$r(n+1)=\max\{r(n+1),\alpha \times C\} \tag{3}$$

where $\alpha$ is a coefficient which is a bit larger than 1. In next section, $\alpha$ used is 1.1. Then, an average fair share of the available capacity for per active ABR VC, which is defined as active during the $n$th sampling period if at least one of its cells arrived at the switch, is computed as follows:

$$\text{Fair\_share}=\frac{r(n+1)\text{\_limited\_VC\_Capacity}}{\text{active\_VCs\_num}-\text{limited\_VCs\_num}} \tag{4}$$

where active_VCs_num is the total number of active ABR VCs, limited_VCs_num is the number of limited VCs which cannot use all their fair share of the capacity because of a limited PCR or a bottleneck somewhere else along their path, and limited_VC_Capacity is the sum of limited VCs capacity.

When a backward RM cell arrives at the switch, the explicit rate for the BRM cell's corresponding VC is calculated as

$$ER=\min\{ER,\text{Fair\_share}\} \tag{5}$$

## 3 Simulation

### 3.1 Simulation model

#### 3.1.1 Network models

In this section, we present some simulation results to verify the performance of the fuzzy mechanism. Two network models are considered. The single bottleneck link model, shown in Fig.6, contains five ABR VCs and an aggregated VBR VC sharing a bottleneck link.

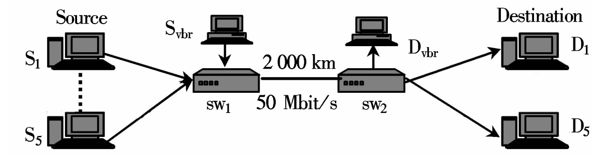



**Fig.6** Single link network simulation model

It is used to investigate the stability, the robusticity in the sense that the mechanism's stability is not sensitive to the change in the number of active VCs, the interoperability between ABR feedback control and the rate variation of VBR VC and their impact on the queue length of the bottleneck switch such as $sw_1$. The parking lot network model is used to investigate the fairness of the mechanism as shown in Fig.7.

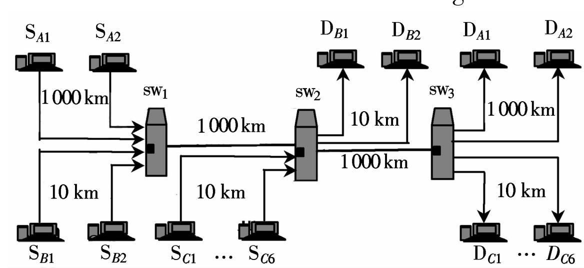



**Fig.7** Parking lot network simulation model

In Fig.6 and Fig.7, $S_{xn}$ ($x$ = blank, $A$, $B$, $C$ and $n=1,2,\cdots,6$) denotes an ABR SES, and $D_{xn}$ is its DES. In Fig.6, all links between the sources/destinations and switches are at a distance of 10 km and have a speed of 149.79 Mbit/s, and the backbone link are set as shown in the figure. All links in Fig.7 have a link rate of 150 Mbit/s.

#### 3.1.2 Source models

Several types of traffic sources are used in the simulations. For ABR sources, when the robusticity is investigated, the characteristics of five ABR sources in Fig.6 are shown in Fig.8, and in other simulations we use greedy sources such that each source generates constant bit rate traffic that fills up as much bandwidth as available to the source.

The values of major parameters used for all ABR end systems in Fig.6 are shown in Tab.2. The values of parameters for all ABR end system in Fig.7 are similar to ones in Tab.2 and the exclusive difference is that PCR of the end system in Fig.7 is 200 Mbit/s.

**Tab.2** ABR end system parameters

| Parameter | NRM | ICR/(Mbit · s$^{-1}$) | PCR/(Mbit · s$^{-1}$) | MCR/(Mbit · s$^{-1}$) |
|---|---|---|---|---|
| Value | 32 | 4 | 149.79 | 1.49 |

In Fig.6, VBR traffic source is the ON-OFF

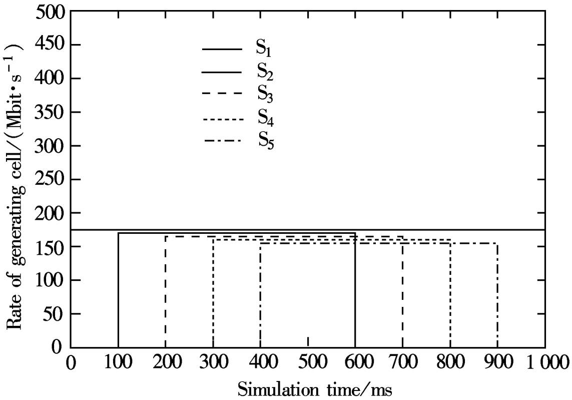



**Fig**.8 The characteristics of five ABR sources in robustcity simulation

process that is a poisson stochastic process. Cells are generated at the specified bit rate that is 20 Mbit/s during the ON state and 0 during the OFF state. Mean ON and OFF state durations are 2.5 ms each, and this results in a mean rate of 10 Mbit/s. The actual periods of ON and OFF state are drawn from an exponential distribution.

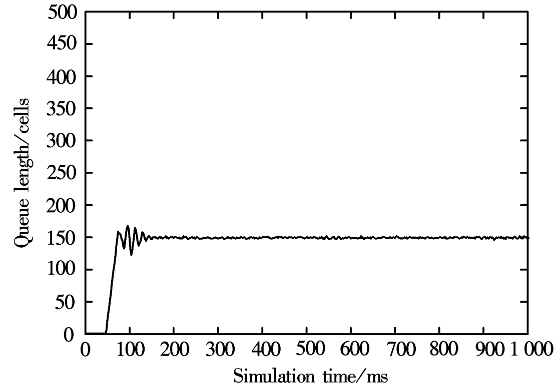
## 3.2 Results

### 3.2.1 Stability

Since the fuzzy mechanism can regulate the rate of ABR SES over time, we are particularly interested in analyzing both the transient and steady behaviors of the network. Transient behaviors such as the duration of response time and the maximum transient queue length are usually the main interests in performance analysis[10].

In this simulation, the network model used is shown in Fig.6. We consider the case where the VBR VC is off, hence the available capacity and the number of active ABR VCs are both fixed. Fig.9 shows the dynamics of the queue length of the bottleneck switch $sw_1$ and ACR of an ABR SES such as $S_1$.

Since the buffer of $sw_1$ is empty at the beginning, the five ABR SESs send out the cells at their maximum allocated rates. Since the total input rate to the switch exceeds the switch's capacity, the queue length increases rapidly. After the information on ER related to queue length is sent back to the SESs, the transmission rate at each SES is quickly reduced. From Fig.9(a), we know that the queue length converges to its target level QT within about 66 ms and its maximum overshoot is 18 cells (12%) at 48 ms. Fig.9(b) shows ACR of $S_1$ converges to 10 Mbit/s which is one fifth of the total capacity of the bottleneck link at 68 ms. The total ACR of five ABR SESs at the steady rate is 50



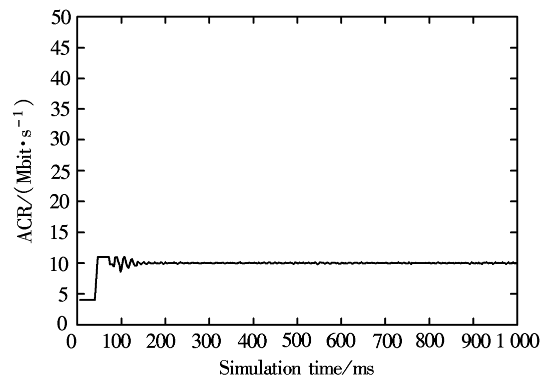




**Fig**.9 Stability simulation: VBR VC is off. (a) Queue length dynamics; (b) ACR dynamics

Mbit/s, which is just equal to the capacity of the bottleneck link. Thus the mechanism has achieved a high utilization of the link bandwidth. Otherwise, from Fig.9 it is shown that the mechanism exhibits low oscillation. From above, we can conclude that our mechanism is stable and has the excellent performance of transient and steady state.

### 3.2.2 Robusticity

Now we study how the mechanism responses to the change of active_VCs_num. Like above, we assume that the VBR VC is off. Let the number of the active VCs increase first from one to five and decrease afterwards from five to one as observed in Fig.8. Fig. 10(a) shows the dynamics of the queue length of $sw_1$, and Fig.10(b) shows the ACR dynamics of $S_1$. It can be seen from Fig.10 that the mechanism responses quickly to the change of active_VCs_num. From Fig. 10(b), we know that the ACR is decreased if active_VCs_num increases, and the ACR is increased if active_VCs_num decreases. From Fig.10(a), it can be seen that the queue length is piecewise stable and always converges to the target level QT in a limited time. In conclusion, the mechanism is robust.

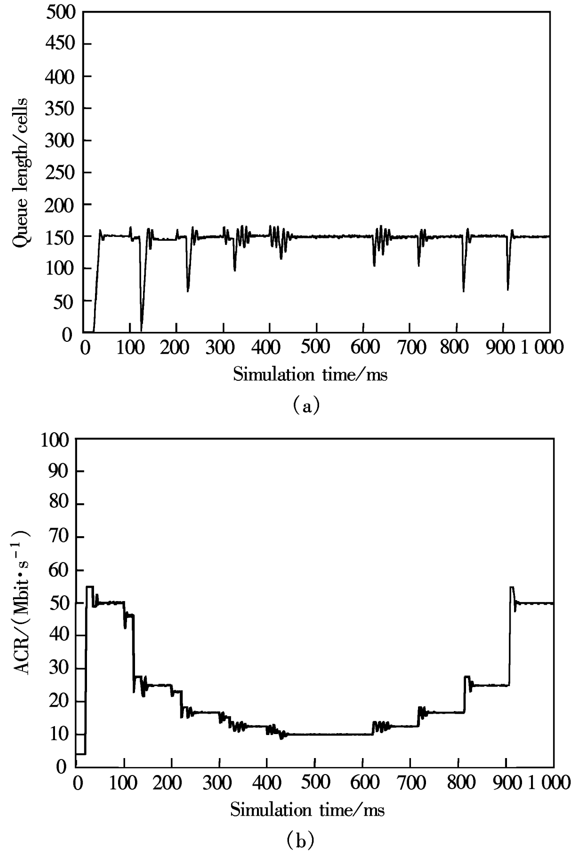



**Fig**. 10 Robusticity Simulation: VBR VC is off. (a) Queue length dynamics; (b) ACR dynamics

### 3.2.3 Interoperability

Now we study the responses of the queue length of the bottleneck switch and ACR of an ABR SES when the available capacity changes. We feed greedy ABR VCs and an ON-OFF VBR VC into the bottleneck link. ACR of $S_1$ and the queue length of $sw_1$ are shown in Fig. 11. Fig. 11(b) shows the ACR oscillates around its steady state value 8 Mbit/s which is equal to (50 − 10)/5 Mbit/s. Like ACR, the queue length oscillates around QT shown in Fig. 11(a). Since the queue length is controlled around the target level QT, the available capacity is fully utilized.

### 3.2.4 Fairness

The primary goal in this experiment is to examine the fairness of our mechanism. We use the ACR dynamics to get insight on fairness. From Fig. 12(a) we note that the tight control on the queue length causes the ACR to converge to the expected fair share for the various VC groups shown in Fig. 12(b). Fairness among VCs of the same group is achieved although due to the lack of space we only show the rate of one VC from each group.

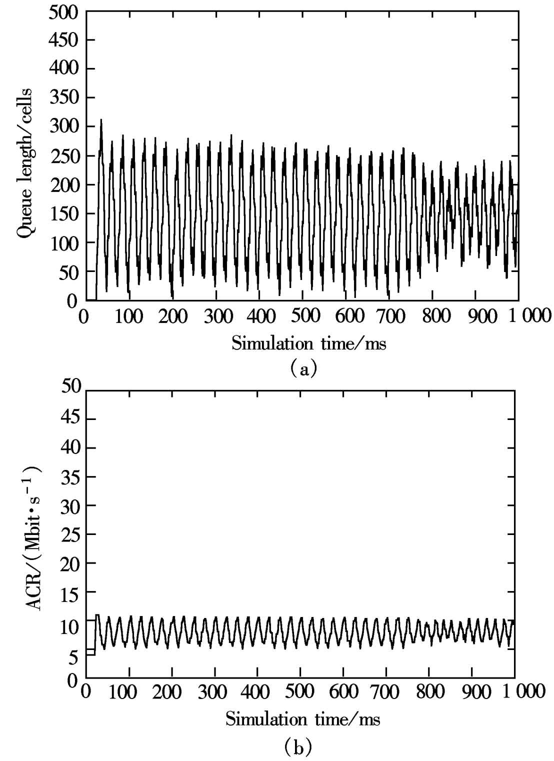



**Fig**. 11 Interoperability simulation. (a) Queue length dynamics; (b) ACR dynamics

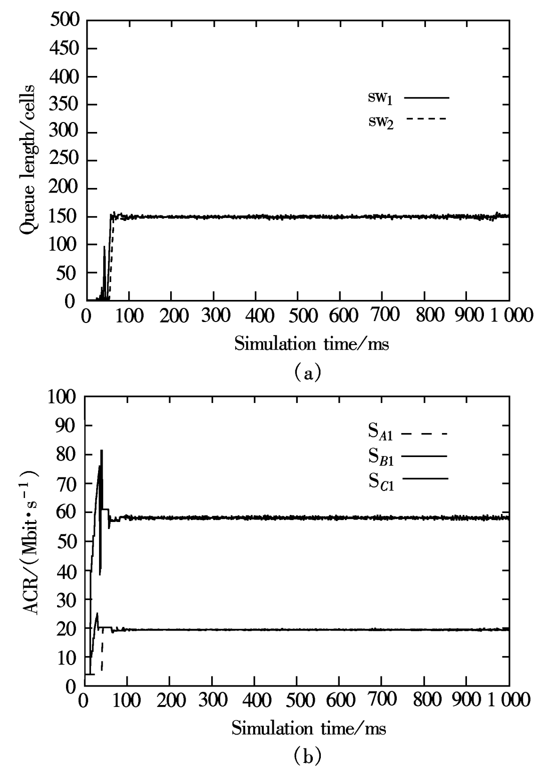



**Fig**. 12 Fairness simulation. (a) Queue length dynamics; (b) ACR dynamics

## 4 Conclusion

This paper presents a formal method of designing an ABR flow control mechanism for ATM networks that can accommodate both guaranteed service and best-effort service traffic. Based on a fuzzy logic controller, this method achieves a stable mechanism. We show how the method is used to design the explicit rate control mechanism. Several numerical simulations are presented to show the stability, fairness, the high utilization of link bandwidth, the excellent transient and steady state performance, and the robusticity in the sense that the mechanism is insensitive to the change of number of active VCs during the network operation.

## References


[1] ATM Forum. Traffic management specification version 4.1 [EB/OL]. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121.000.pdf. 1999-03.

[2] Kamolphiwong S, Karbowiak A E, Mehrpour H. Flow control in ATM networks: a survey[J]. *Computer communication*, 1998, **21**:951-968.

[3] Makrucki B A. Explicit forward congest notification in ATM networks[A]. In: *Proc. of TriComm*'92[C]. New York, 1992.73.

[4] Bonomi F, Fendick K W. The rate-based flow control framework for the available bit rate ATM service[J]. *IEEE Network*, 1995, **9**:25-39.

[5] Rohrs C E, Berry R A, O'Halek D. A control engineer's look at ATM congest avoidance[J]. *Computer communication*, 1996, **19**:226-234.

[6] Su C F, Veciana G D, Walrand J. Explicit Rate Flow Control for ABR Services in ATM Networks[J]. *IEEE/ACM Transactions on networking*, 2000, **8**:350-361.

[7] Pitsillides A, Sekercioglu Y A, Ramamurthy G. Effective control of traffic flow in ATM networks using Fuzzy explicit rate marking (FERM)[J]. *IEEE journal on selected aeras communications*, 1997, **15**(2):209-225.

[8] Mohamad Hossien, Mostafa S, Mohamad Bagher M. Intelligent usage parameter controller based on dynamic rate leaky bucket for ATM networks[J]. *Computer Networks*, 2000, **32**(1):17-34.

[9] Nelson O L, Tham C K. Connection admission control of ATM network using integrated MLP and fuzzy controllers[J]. *Computer Networks*, 2000, **32**(1): 61-79.

[10] Golmie N, Saintillan Y, Su D. ABR switch mechanisms: design issues and performance evaluation[J]. *Computer networks and ISDN systems*, 1998, **30**:1749-1761.


# ATM 网络中面向 ABR 服务的一种模糊流量控制方法


张孝林　张飒兵　吴介一

（东南大学 CIMS 中心，南京 210096）



**摘　要**　面向 ABR 服务的显式速率流量控制机制用于公平和合理地分配瓶颈链路可提供的带宽，保持与瓶颈链路连接的交换机缓冲区队列长度在所希望的程度，从而可以预防和控制 ATM 网络中的拥塞.然而，设计有效的流量控制机制较为困难，这主要是由于网络中的动态参数，如可提供的链路带宽、业务量的并发以及 ABR 源-目的之间距离等变化的影响.对此，本文提出了一种面向 ABR 服务的模糊显式速率控制机制，该机制结构简单，鲁棒性强.仿真实验表明：该机制可以有效地避免网络拥塞，公平地分配链路带宽并且具有快速收敛、低振荡、高链路带宽利用率的特点.



**关键词**　ATM 网络，拥塞预防，流量控制，模糊逻辑

**中图分类号**　TP393