To appear in Proc. of IEEE INFOCOM'97.

Integrated Rate and Credit Feedback Control for ABR Service in ATM Networks

Xi Zhang and Kang G. Shin

Real-Time Computing Laboratory
Dept. of Elec. Engin. and Compt. Scie.
The University of Michigan
Ann Arbor, MI 48109–2122
{xizhang,kgshin}@eecs.umich.edu

Qin Zheng

Mitsubishi Electric Research Labs
Cambridge Research Center
201 Broadway
Cambridge, MA 02139
zheng@merl.com

Abstract

We propose a flow-control scheme that combines the merits of credit- and rate-based flow-control schemes by applying direct control over both bandwidth and buffer resources. The goal of the proposed scheme is to design an optimal rate-control policy for a given ***finite*** *buffer capacity that maximizes average throughput and bounds end-to-end delay. By applying higher-order rate control, the proposed scheme not only makes the rate process converge to the neighborhood of link bandwidth, but also confines the queue-length fluctuation to a regime bounded by buffer capacity (thus guaranteeing lossless transmission). Using the fluid approximation method, we model the proposed flow-control scheme and study the system dynamic behavior for ABR (Available Bit Rate) service under the most stressful traffic condition. We derive the expressions for queue build-ups and average throughput in both transient and equilibrium states. The analytical results have shown the proposed scheme to be stable and efficient in that the source rate and bottleneck queue length rapidly converge to the designated operating region. Also, presented are examples showing that the proposed scheme outperforms the other existing schemes.*

1 Introduction

An ATM network can transport a wide variety of information such as data, audio, and video. Different types of user traffic have different requirements on bandwidth, loss ratio, and delay, which are characterized by a set of traffic parameters. Based on these traffic parameters, the ATM network sets up a connection (or VC—Virtual Circuit) from the source to the destination. A connection runs through a series of intermediate switch nodes, where it shares link bandwidth and buffer space with other connections. Thus, the traffic rate flowing through a switch depends on the number of connections and the source rates of these connections. To achieve high bandwidth utilization in the face of bursty traffic, the connections sharing the same output link are statistically multiplexed at the switch. However, if all of these connections become active simultaneously, or some connections increase their rates unlimitedly, queues build up at bottle-necked switches. Eventually, the buffer capacity is exceeded and cells are dropped, resulting in low throughput, a large delay, and even network blockage. To prevent a network from falling into this kind of congestion, an efficient flow-control scheme is required.

Available Bit Rate (ABR) service, which is suitable for various data communications, can maximize network bandwidth utilization and avoid congestion. In ABR service, there is no strictly-specified contract between the network and a client that describes the traffic behavior and the expected quality of service. Rather, the network is expected to provide each client with a fair share of available bandwidth dynamically; so ABR is a best-effort service. After allocating a certain bandwidth to high-priority traffic, such as Constant Bit Rate (CBR) connections, the network divides the remaining bandwidth among ABR connections. The client should also adjust his transmission rate based on the feedback on network congestion. So, ABR service requires a closed-loop congestion-control scheme, dynamically regulating the cell-transmission rate of each source according to congestion status.

A number of flow-control schemes have been proposed for ABR service. Among these, both *credit* [1] and *rate* [2,3] schemes have received most attention [4]. The credit scheme guarantees lossless transmission by applying direct control over buffer space for each connection in a hop-by-hop manner. However, the credit scheme cannot make a bandwidth guarantee for each connection since it is window-type flow control and does not regulate the traffic flow rate [5]. Moreover, the credit scheme attempts to keep the buffer full to achieve high utilization. This may result in unbounded end-to-end delays and large delay variations. In contrast, the rate scheme provides a bandwidth guarantee and a bounded delay to each connection by exercising direct control over the link bandwidth allocated to each connection in an end-to-end fashion. But the buffer requirement for the rate scheme is very large and increases with feedback delay, the number of active connections,

The work reported in this paper was supported in part by a grant from Mitsubishi Electric Research Center, Cambridge, MA, and by the ONR under Grant N00014-94-0291. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

and the initial rate [6–9]. This makes the buffer design very difficult since the exact value of each connection's feedback delay and the number of active connections over a given link are not known *a priori*.

The aforementioned problems with the credit and rate schemes stem from the fact that neither scheme exerts direct control over both link bandwidth and buffer space. In this paper, we propose an integrated flow-control scheme that applies direct control over *both* link bandwidth and buffer space, to achieve the following goals:

- Lossless transmission for given finite buffer capacity,
- Optimal rate control to maximize average throughput for given buffer capacity,
- Bounded end-to-end delay,
- Fair bandwidth share guaranteed among competing connections,
- Maximum network utilization.

Using the first-order fluid approximation method[2, 10], we model the proposed scheme and analyze the system's dynamic behavior for ABR service under the most stringent traffic condition. In previous performance analyses, the maximum queue length Q_{max} was treated as a free parameter under the unrealistic assumption of infinite buffer capacity [2,6–11]. In contrast, we assume the buffer capacity C_{max} is *finite* and use $Q_{max} < C_{max}$ as a constraint to find the optimal rate-control function. We derive closed-form expressions to evaluate the scheme's performance and compute the evolutions of rate and queue functions for transient and equilibrium states. From the analysis, we identify the optimal control pattern/state and conclude that just exercising increase/decrease rate control cannot have the system converge to the optimal control state (specified by bandwidth and buffer allocations). A higher-order rate control is applied over the rate-increase parameter with an exponential decreasing rule. Applying a two-dimensional rate control in the transient state analysis shows that the system rapidly converges to the designated optimal operating regime.

This paper is organized as follows. In Section 2, we compare the rate and credit schemes, and identify the problems with them. Section 3 presents our proposed scheme to solve these problems. Section 4 deals with the system model and the control model for the proposed scheme. In Section 5, we derive analytical solutions for both transient and equilibrium states and evaluate the scheme's performance for the single-connection case. Section 6 analyzes the proposed scheme's performance for the multiple-connection case through examples. The paper concludes with Section 7.

2 Rate vs. Credit, and Interworking

The principles and control mechanisms of the rate and credit schemes are detailed in [1] and [3]. Here, we focus on comparing them in terms of structures and performance and arguing for the need to integrate them.

The rate scheme regulates a connection's bandwidth by directly controlling its source cell-transmission rate according to network congestion information. Using RM (Resource Management) cells and EFCI (Explicit Forward Congestion Indication) bit setting, the information feedback control loop spans the entire network in an end-to-end fashion. The rate scheme aims at providing a bandwidth guarantee to each VC, bounding end-to-end transmission delay, and achieving fair allocation of network resources. On the other hand, the credit scheme exercises direct control and feedback on the amount of space left in switch buffers, rather than the rate. Instead of exercising an end-to-end control algorithm, the credit scheme segments the control loop at each switch. The goal of credit scheme is to ensure lossless transmission with a given finite buffer capacity while maintaining high bandwidth and buffer utilization.

Depending on their different goals and structures, these two schemes each have their own advantages and disadvantages, which will be discussed below.

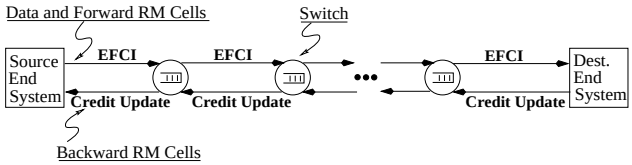
Lossless transmission and buffer requirement: With the rate scheme, the buffer requirement is very large and increases with feedback delay, the number of active connections, and the initial rate [6–9]. This makes buffer design very difficult, because the exact values of the network delay of each connection and the number of active connections over a given link are not known *a priori*. As a result, one is forced to compromise between buffer size and loss ratio. In contrast, the credit scheme supports lossless transmission for any given finite buffer size.

Bandwidth guarantee: By explicitly assigning a target bandwidth to each connection, the rate scheme is most suitable for bandwidth-guaranteed applications. The credit scheme, like other window-type flow-control schemes, does not provide any bandwidth guarantee to each connection since it does not directly regulate the transmission rate.

End-to-end delay and delay variation: In the credit scheme, trying to always keep the buffer full may lead to larger end-to-end delays and delay variations. On the other hand, the rate scheme guarantees bandwidth for each VC and thus, each VC can receive guaranteed throughput. So, shaping traffic for each VC allows the end-to-end delay to be bounded.

Network resource utilization: Using a hop-by-hop feedback protocol, the credit scheme tends to achieve very high network utilization even in the face of widely-varying traffic loads, because buffered data can be sent whenever such an opportunity arises. But for the rate scheme, it is difficult to achieve high utilization of bandwidth due to large end-to-end delay. Moreover, if lossless or low-loss transmission is required, a very large buffer must be provided at each switch. This large buffer may be severely underutilized when only a small portion of VCs are active. By contrast, the credit scheme can ensure lossless transmission with a much smaller buffer while keeping it highly utilized.

Flow control is basically a resource management and control problem in a shared and distributed network environment. Network resources are composed of link bandwidth and buffer space. However, neither of the two schemes exerts direct control over both of these resources. Thus, an efficient flow-control scheme should



RM Cell Format:

DIR	0=Forward; 1=Backward
BN	0=Source Generated; 1= Switch Generated
CI	0=Not Congested; 1=Congested
ER	Explicit Cell Rate (two bytes)
CCR	Current Cell Rate (two bytes)
MCR	Minium Cell Rate (two bytes)
CU	Credit Update Value (two bytes)

Figure 1: Basic framework and RM cell format of the proposed scheme.

apply *direct* control over *both* bandwidth and buffer resources.

3 The Proposed Scheme

Observing the complementary features of the rate and credit schemes, we propose an *integrated* flow-control scheme which combines their merits while overcoming their drawbacks.

3.1 Key Differences from Rate or Credit Scheme

The framework and RM cell format for the proposed scheme are illustrated in Figure 1. Our scheme also uses the EFCI bit and RM cell to convey network congestion information. The EFCI bit is used for rate control and the backward RM cell is used for updating credit balance. Here the RM cell is redefined such that it contains both rate and credit control information. In particular, we added a new CU (Credit Update) field in the RM cell and use the BN (Backward Notification) bit to distinguish the RM cells generated either by the source or by intermediate switch nodes. Both rate and credit control are applied at all nodes using the redefined RM cells. Our scheme discriminates between two types of congestion: (1) *bandwidth congestion*, if queue length $Q(t) > Q_h$, a threshold; (2) *buffer congestion*, if credit balance $C_{bal} = 0$. If a buffer congestion occurs at a switch, the switch sends a backward RM cell (with BN=1) back to the source for a quick release of buffer congestion. There are two rate control modes at the source corresponding to these two types of congestion: (i) if a bandwidth congestion occurs then the source rate is reduced exponentially from its current value; (ii) if a buffer congestion occurs then the source needs to:

- cut down its current ACR (Allowed Cell Rate) to an appropriate smaller value R_c, which is less than the bottleneck bandwidth μ, but larger than its MCR (Minimum Cell Rate);
- exponentially reduce the rate-increase parameter which is the second-order rate control.

These enhanced features in structures and algorithms enable the proposed scheme to cope with the following practical problems that the other two schemes cannot handle. For given buffer capacity, our scheme adaptively adjusts rate-control parameters such that the system can quickly converge to an optimal rate-control mode, which maximizes average throughput, guarantees lossless transmission, and lowers end-to-end delay. On the other hand, when an established ABR connection specifies its MCR, ICR (Initial Cell Rate) and the corresponding rate control parameters, the proposed scheme can provide information on the optimal buffer allocation for each connection to meet its performance specifications.

3.2 The Control Algorithms

The control algorithms are involved with the Source End System (SES), the Destination End System (DES), and all Intermediate Switch Systems (ISS) between SES and DES.

The Source Node Algorithm (Table 1). SES deals mainly with two events: sending data cells (lines 03–12) and receiving RM cells (lines 13–28). When the rate-control timer expires, SES first check if credit balance C_{bal} is positive. If $C_{bal} > 0$, it sends a data cell to the downstream node and then increments the count and decrements the credit (book-keeping). SES sends an RM cell once every N_{rm} data cells. Then the rate-control timer is reset. Upon receiving an RM cell, SES first updates its local credit balance by CU contained in the RM cell and then starts rate control. SES first check if the VC is already in buffer congestion state. If

```
00. Local Variables: ACR := ICR; U_cnt := 0; C_bal := C_max;
01. Buffer_congestion := 0; Data_que_len := 0; RM_send := 0;
02. while (VC_on_line) {
03.   if (Current_time ≥ Next_cell_time)  ! Sending data event
04.     if (C_bal > 0 and Data_que_len > 0)
05.       send data cell with EFCI := 0;
06.       Data_que_len := Data_que_len − 1;
07.       U_cnt := U_cnt + 1;   C_bal := C_bal − 1;   ! Book-keeping
08.     if (RM_send Mod N_rm = 0)
09.       send RM(DIR := forward, CI := 0, CCR := ACR,
10.               MCR, ER := PCR, BN := 0) cell;
11.     RM_send := RM_send + 1;
12.     Next_cell_time := Next_cell_time + 1/ACR;
13.   if (receive RM(DIR = backward, CI, CCR, ER,
14.                  CU = D_cnt, BN) cell)
15.     C_bal := C_max − (U_cnt − CU);
16.     if (Buffer_congestion = 0 or CI = 0)
17.       if (BN = 1)
18.         ACR := max{(ER/ACR)(2 * ER − ACR), MCR};
19.         AIR := 0.5AIR;     ! AIR (Additive Increase Rate)
20.         MDF := e^(−AIR/ER);
21.         Buffer_congestion := 1;
22.       else if (CI = 1)
23.               ACR := ACR * MDF;
24.               if (C_bal = 0)  ACR := MCR;
25.             else
26.               ACR := ACR + AIR;
27.               Buffer_congestion := 0;
28.       Next_cell_time := Current_time + 1/ACR;
29. }
```

Table 1: Pseudocode for SES.

no buffer congestion, then (1) if $BN = 0$ then SES *additively* increases its ACR or *multiplicatively* decreases its ACR depending on the CI bit (set ACR to MCR if C_{bal}=0); (2) If $BN = 1$ then SES turns into buffer congestion state, and exercises the buffer congetion control by setting ACR to R_c, exponentially reducing rate-increase parameter AIR and accordingly adjusting rate-decrease parameter MDF. If this VC is already in the buffer congestion state, then the source ACR stays with R_c until the first backward RM cell with CI=0 (non-

congestion) is received. Finally, the rate-control timer is adjusted according to the updated ACR (for simplicity only the per-N_{rm} data cells scheme is presented. Our scheme also allows periodic rate-update control.)

The Switch Node Algorithm (Table 2). Three main events need to be handled: (1)*Receiving data* (lines 02–11): forward the data cell if the output link is ready and $C_{bal} > 0$; enqueue the data otherwise. Mark the

```
00. Local Variables: U_cnt, C_bal, Data_que, Data_que_len,
01. Local_VC_CCR; Local_VC_ER; Local_VC_CI := 0;
02. if (receive Data cell)      ! Receiving data event
03.   if (Output link is ready and C_bal > 0)
04.     forward Data cell;
05.     U_cnt := U_cnt + 1;  C_bal := C_bal - 1;  ! Book-keeping
06.   else
07.     add Data cell to Data_que;
08.     if (C_bal = 0)
09.       send RM(DIR := backward, CU := U_cnt, CI := 1,
10.                                ER := μ, BN := 1)
11.     if (Data_que_len ≥ Q_h)  Local_VC_CI := 1;
12. if (receive output link ready signal and Data_que_len > 0 )
13.   schedule all active VCs {
14.   if (C_bal > 0)
15.     remove Data cell from head of Data_que;
16.     if (Data_que_len ≤ Q_l)  Local_VC_CI := 0;
17.     forward Data(EFCI := EFCI ⊕ Local_VC_CI) cell;
18.     U_cnt := U_cnt + 1;    C_bal := C_bal - 1;  ! Book-keeping
19.   else
20.     send RM(DIR := backward, CU := U_cnt, CI := 1,
21.                          ER := μ, BN := 1) cell }
22. if ( receive RM(DIR, CI, CCR, ER, CU, BN) cell )
23.   if (DIR = forward)
24.     Local_VC_CCR := CCR;    Local_VC_ER := ER;
25.     send RM(DIR = forward) cell;
26.   else
27.     C_bal := C_max - (U_cnt - CU);          ! CU = D_cnt
28.     send RM(DIR = backward, CU := U_cnt,
29.                          CI := CI ⊕ Local_VC_CI) cell;
```

Table 2: Pseudocode for ISS.

local CI bit ($Local\_VC\_CI$) for setting EFCI bit in the data cell header if the queue length exceeds Q_h. Generate and send an RM cell directly back to the source with $BN = 1$, $CI = 1$, and $ER =$ link bandwidth if $C_{bal} = 0$ after credit book-keeping. (2)*Receiving link-ready signal* (lines 12–21): schedule the active VCs. Dequeue a data cell for the scheduled VC. If queue length drops below Q_l, the local CI bit is unmarked. Generate and send an RM cell directly back to SES with BN=1, $CI = 1$, and $ER =$ bandwidth if $C_{bal} = 0$. (3)*Receiving RM cells* (lines 22–29): for a forward RM cell, record its contents and forward it to the downstream node; for a backward RM cell, update the local credit-balance by CU contained in the RM cell, fill in the RM cell with local count and CI bit, and then send it to the upstream node.

The Destination Node Algorithm (Table 3). Two

```
00. Local Variables: Local_VC_CI, U_cnt;
01. if (receive Data cell)  ! Receiving data cell event
02.   Local_VC_CI := EFCI field of Data cell;
03.   U_cnt := U_cnt + 1;
04.   forward Data cell to user;
05. if (receive RM(DIR = forward, CI, CCR, ER, CU, BN))
06.     send RM(DIR = backward, CU := U_cnt,
07.                  CI := CI ⊕ Local_VC_CI, BN := 0 ) cell;
```

Table 3: Pseudocode for DES.

events are processed: receving data cells (lines 01–04)

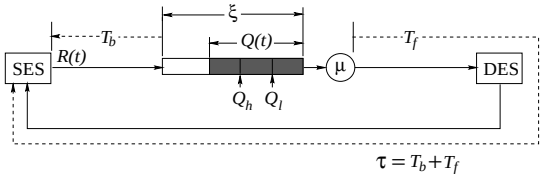


Figure 2: The system model for a virtual circuit.

and receiving RM cells (lines 05–07). When a data cell is received, its EFCI bit is saved and the local count is updated. When an RM cell is received, the RM cell's CI bit is set using the EFCI bit saved from the data cell last received. Finally, return the RM cell with the updated credit and congestion information to the upstream node.

4 System Model

An ATM network with ABR connections subject to the proposed flow-control scheme is a dynamic system. We model this system by using the first-order fluid approximation method [2, 10], where $R(t)$ and $Q(t)$ represent source-rate function and bottleneck queue-length function respectively (see Figure 2). Due to its simplicity, effectiveness, and approximation accuracy (particularly for heavy traffic), the fluid modeling method has been effectively applied to the analysis and evaluation of several common rate-based flow-control schemes [2, 6–11].

In all previous analyses using the fluid model, the maximum queue length Q_{max} is treated as a free parameter under the unrealistic assumption that buffer capacity is infinite [2, 6–11]. In a real network, however, this assumption does not hold, and thus, the results based on this assumption are not applicable to the case of finite buffer capacity. By contrast, our model hinges on a finite buffer capacity C_{max}, and the inequality $Q_{max} < C_{max}$ is used as a constraint in finding the optimal rate-control function. We also assume the existence of only a single bottleneck with queue length $Q(t)$ and a "persistent" source, which always has data cells to send, with ACR = $R(t)$, for each VC. Such a data source model allows us to examine the proposed scheme under the most stressful condition.

4.1 System Description

The system with the proposed flow-control scheme is characterized by the following parameters (see Figure 2):

β:	Multiplicative decrease factor for rate reduction
α:	Additive rate-increase slope
Δ:	Time interval of rate update
MCR:	Minimum cell rate for the ABR connection
Q_h:	High threshold of $Q(t)$ for traffic overload
Q_l:	Low threshold of $Q(t)$ for traffic underload
T_b:	Backward delay of the ABR connection
T_f:	Forward delay of the ABR connection
ξ:	Bottleneck maximum buffer allocation (C_{max})
μ:	Bottleneck link bandwidth (BW)

T_b is the delay experienced by *buffer congestion* signal from the bottleneck to SES. T_b also represents the

delay for $R(t)$'s action to reach the bottleneck from SES. T_f is measured from the detection of *bandwidth congestion* at the bottleneck to the time when an EFCI signal reaches SES via DES. Thus, $\tau = T_b + T_f$ is the VC's round-trip delay (including processing time and propagation delay.) We use the synchronous model for rate control in which the fixed (periodic) rate-update interval Δ is usually a fraction of τ.

The additive increase and the multiplicative decrease of rate control during the n-th rate-update interval are expressed as:

$$R_n = \begin{cases} R_{n-1} + a; & \text{Add. increase, } a = AIR \\ bR_{n-1}; & \text{Multi. decrease, } b = MDF \end{cases} \quad (4.1)$$

4.2 System State Equations

The system state is specified by two state variables: $R(t)$ and $Q(t)$. According to the proposed control algorithms, the system state equations for a VC containing a single bottleneck are given by the following equations, depending on whether rate or credit control is in operation.

Rate-control: For $C_{bal} > 0$ (no-buffer congestion)

$$R(t) = \begin{cases} R(t_0) + \alpha(t - t_0); & \text{If } Q(t - T_b) < Q_h \\ R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}}; & \text{If } Q(t - T_b) \geq Q_h \end{cases} \quad (4.2)$$

$$Q(t) = \int_{t_0}^{t} [R(\alpha - T_b) - \mu]d\alpha + Q(t_0). \quad (4.3)$$

where the rate "additive increase" and rate "multiplicative decrease" are modeled by "linear increase" and "exponential decrease", respectively, in a continuous domain [2].

Credit-control: If $C_{bal} = 0$ (buffer congestion)

$$R(t) = R_c, \quad (R_c \geq MCR) \quad (4.4)$$

$$Q(t) = \xi - (\mu - R_c)(t - T_b - t_0). \quad (4.5)$$

where R_c is the cut-down rate set by SES when it receives a (BN=1) RM cell. Note that the non-linear dynamics of the rate functions is due to the fact that $Q(t) \in [0, \xi]$.

5 Analysis of a Single ABR Connection

The system dynamics could be in either equilibrium or transient state, which are treated below separately.

5.1 Equilibrium State Analysis

The equilibrium state is defined as the state in which the source-rate function $R(t)$ and the bottleneck queue-length function $Q(t)$ have already converged to a certain regime and oscillate with constant amplitude and frequency. The use of credit control yields three different patterns for flow-controlled rate and queue-length functions, depending on the range of the flow-control parameters.

Pattern I: $\xi > Q_{max}$. Since $\xi > Q_{max}$, no buffer congestion occurs. The rate-control mechanism governs the system dynamics (see Figure 3).

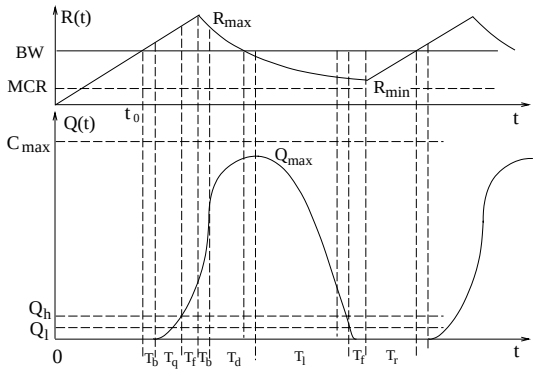


Figure 3: Dynamic Behavior of $R(t)$ and $Q(t)$ for Pattern I.

Let R_{max} be the maximum rate, then we obtain

$$R_{max} = \mu + \alpha(T_q + T_b + T_f) \quad (5.1)$$

where $T_q = \sqrt{2Q_h/\alpha}$ is the time for $Q(t)$ to grow to Q_h from zero. For the convenience of presentation, we define

$$T_{max} \triangleq T_b + T_q + T_f = T_b + \sqrt{\frac{2Q_h}{\alpha}} + T_f \quad (5.2)$$

which is the time for $R(t)$ to increase from μ to R_{max}. Then, we get the maximum queue length

$$Q_{max} = \frac{\alpha}{2}T_{max}^2 + \alpha\frac{\Delta T_{max}}{(1-\beta)} + \frac{\mu\Delta}{(1-\beta)}\ log\ \frac{\mu}{R_{max}}. \quad (5.3)$$

T_l is the duration for $Q(t)$ to decrease from Q_{max} to Q_l, and thus, is determined by the following non-linear equation:

$$e^{-(1-\beta)\frac{T_l}{\Delta}} + \frac{1-\beta}{\Delta}\left(T_l - \frac{Q_{max} - Q_l}{\mu}\right) - 1 = 0. \quad (5.4)$$

The minimum rate is given by

$$R_{min} = \mu e^{-(1-\beta)\frac{(T_l + T_b + T_f)}{\Delta}}. \quad (5.5)$$

Clearly, the rate-fluctuation cycle is $T = T_q + T_d + T_l + 2\tau + T_r$, where $T_d = -\frac{\Delta}{(1-\beta)}\ log\ \frac{\mu}{R_{max}}$ is the time for $R(t)$ to drop from R_{max} back to μ and $T_r = (\mu - R_{min})/\alpha$ is the time for $R(t)$ to grow from R_{min} to μ.

The average throughput in equilibrium state can be calculated by averaging $R(t)$ over one cycle T as

$$\overline{R} \triangleq \frac{1}{T}\int_{t_0}^{t_0+T} R(t)dt = \frac{1}{T}\Bigg[\ \mu T_{max} + \frac{\alpha}{2}T_{max}^2 + \frac{R_{max}\Delta}{1-\beta} \cdot \left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) + T_r R_{min} + \frac{\alpha}{2}T_r^2\Bigg] \quad (5.6)$$

where $T_e = T_d + T_l + \tau$ is the time for exponential-decrease rate control within a cycle.

Pattern II: $\alpha(T_{max} - 2T_b)^2/2 < \xi < Q_{max}$. In this case $Q(t)$ would grow beyond ξ without credit control, but ξ is still large enough for $R(t)$ to reach R_{max}. After $Q(t)$ reaches ξ, the source receives the buffer congestion signal and cuts down $R(t)$ to an appropriate smaller R_c

to achieve a quick dissipation of the buffer congestion. Two factors affect the selection of R_c. If R_c is too large, then the queuing delay increases because the speed of draining a congested buffer is inversely proportional to R_c. On the other hand, if R_c is too small, then the average throughput decreases. To make a tradeoff between queuing delay and average throughput, we set:

$$R_c = \max\left\{ \frac{\mu}{R_{max}}(2\mu - R_{max}),\ \ MCR \right\}. \qquad (5.7)$$

Notice that if $R_{max} \geq 2\mu$ then $R_c = MCR$; and if $R_{max} = \mu$ then $R_c = \mu$.

The rate control in Pattern II is further divided into three cases because they need different analytical treatments. For convenience of presentation, we introduce a parameter T_c, the time for $Q(t)$ to increase from 0 to ξ. The system dynamics belong to one of these three cases, depending on the range T_c falls in.

Case 1: $\underline{T_{max} - 2T_b < T_c < T_{max}.}$ We get $T_c = \sqrt{2\xi/\alpha}$. The next key parameter is T_l, which is the time between $Q(t)$ reaching ξ and $Q(t)$ dropping to Q_l, and given by

$$T_l \ = \ 2T_b + \frac{\theta + (\xi - Q_l)}{\mu - R_c} \qquad (5.8)$$

where

$$\theta = \frac{\alpha}{2}T_{max}^2 + \frac{\Delta R_{max}}{1-\beta}\left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) - \mu T_e - \xi \quad (5.9)$$

is the number of "overshoot" cells the bottleneck node cannot accept due to $Q(t) = \xi$ and have been temporarily saved by the previous nodes. In Eq. (5.9) $T_e = (T_c + 2T_b) - T_{max}$. Then, we get $T = T_c + T_b + T_l + T_f + (\mu - R_c)/\alpha$.

By the definition of $\overline{R}$ given in Eq. (5.6), we get

$$\overline{R} = \frac{1}{T}\Bigg[\mu T_{max} + \frac{\alpha}{2}T_{max}^2 + \frac{\Delta R_{max}}{1-\beta}\left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) + (T_l - T_b + T_f)R_c + T_r R_c + \frac{\alpha}{2}T_r^2 \Bigg] \qquad (5.10)$$

Case 2: $\underline{T_{max} < T_c < T_{max} + T_d - 2T_b.}$ Since $T_c > T_{max}$, T_c consists of two parts, $T_c = T_{max} + T_\xi$ where T_ξ is the duration for the exponential part of $R(t)$ contributing toward $Q(t) = \xi$ and is determined by a nonlinear equation:

$$e^{-(1-\beta)\frac{T_\xi}{\Delta}} + \frac{1-\beta}{\Delta}\left(\frac{\xi - \frac{\alpha}{2}T_{max}^2}{R_{max}} + \frac{\mu T_\xi}{R_{max}}\right) - 1 = 0. \qquad (5.11)$$

The expressions of T_l, θ, T, and $\overline{R}$ for Case 2 are the same as Case 1 except that in Case 2, $T_c = T_{max} + T_\xi$ and $T_e = T_\xi + 2T_b$.

Case 3: $\underline{T_{max} + T_d - 2T_b < T_c < T_{max} + T_d.}$ Here T_l is a function of the "net" contribution of the "overshoot" cells: $\theta + \gamma$ where θ (γ) denotes the positive (negative) contribution generated in a delay interval of $2T_b$ when $R(t)$ curve intersects μ and is given by

$$\theta = \frac{\alpha}{2}T_{max}^2 + \frac{\Delta R_{max}}{1-\beta}\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) - \mu T_d - \xi \quad (5.12)$$

$$\gamma = \mu\left[\frac{\Delta}{1-\beta}\left(1 - e^{-(1-\beta)\frac{2T_b - (T_d - T_\xi)}{\Delta}}\right) - 2T_b + T_d - T_\xi\right] \qquad (5.13)$$

Computing T as in Case 1, except that here

$$T_l = 2T_b + \frac{(\theta + \gamma) + (\xi - Q_l)}{\mu - R_c}, \qquad (5.14)$$

we get the average throughput for Case 3 as follows:

$$\overline{R} = \frac{1}{T}\Bigg[\mu T_{max} + \frac{\alpha}{2}T_{max}^2 + \frac{\Delta R_{max}}{1-\beta}\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) + \frac{\Delta\mu}{1-\beta}\left(1 - e^{-(1-\beta)\frac{2T_b - (T_d - T_\xi)}{\Delta}}\right) + (T_l - T_b + T_f)R_c + T_r R_c + \frac{\alpha}{2}T_r^2 \Bigg]. \qquad (5.15)$$

Pattern III: $\underline{0 < \xi < \alpha(T_{max} - 2T_b)^2/2.}$ In this pattern, $T_c = \sqrt{2\xi/\alpha}$ and $R_{max} = \mu + \alpha(T_c + 2T_b)$. Using the same definitions of θ, T_l, and Eq. (5.8) as in Pattern II, we obtain

$$\theta \ = \ \frac{\alpha}{2}(T_c + 2T_b)^2 - \xi, \qquad (5.16)$$

$$T_l \ = \ 2T_b + \frac{\frac{\alpha}{2}(T_c + 2T_b)^2 - Q_l}{\mu - R_c}. \qquad (5.17)$$

Substituting T_c and T_l into T given in Pattern II and using (5.6), we have Pattern III's $\overline{R}$ as given below:

$$\overline{R} = \frac{1}{T}\Bigg[\mu(T_c + 2T_b) + \frac{\alpha}{2}(T_c + 2T_b)^2 + (T_l - T_b + T_f)R_c + T_r R_c + \frac{\alpha}{2}T_r^2 \Bigg]. \qquad (5.18)$$

5.2 Numerical Evaluation of Equilibrium-State Performance

We set the bottleneck link bandwidth $\mu = 155$ Mbps and we assume $T_b = T_f = 1$ ms and hence, $\tau = T_b + T_f = 2$ ms. Also, we use $\Delta = 0.5\tau = 1$ ms, $Q_h = 50$ cells, $Q_l = 25$ cells, and the initial source rate $R_0 = \mu$. To balance the increase and decrease speeds of $R(t)$ and ensure that the average of the offered traffic load does not grow beyond the bottleneck bandwidth, we set $\alpha\Delta/(1-\beta) = \mu$ [10] throughout the rest of the paper. In the following we present some of the numerical results we obtained to evaluate system performance and the more complete and detailed results can be found in [12].

Performance Analysis for Pattern I, II, and III: As expected, Figure 4 shows Q_{max} increases monotonically with α and τ. Q_{max} also increases roughly linearly with α and Q_{max} increases faster for a larger τ. In Figure 5, $\overline{R}$ is found to decrease monotonically as α and τ increase, and to decrease faster for a larger τ. In general, a large τ has a negative effect on equilibrium-state performance, which is consistent with feedback system analysis. A small α is desired for equilibrium-state performance in terms of the maximum queue length and

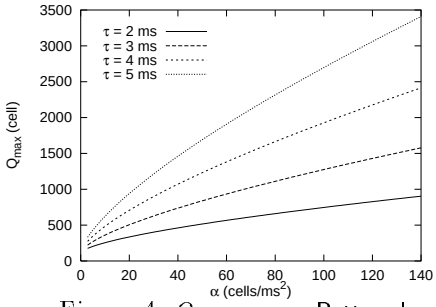


Figure 4: Q_{max} vs. α: Pattern I

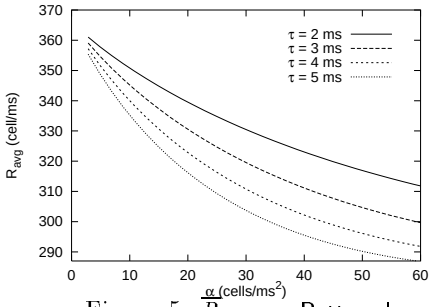


Figure 5: $\overline{R}$ vs. α: Pattern I

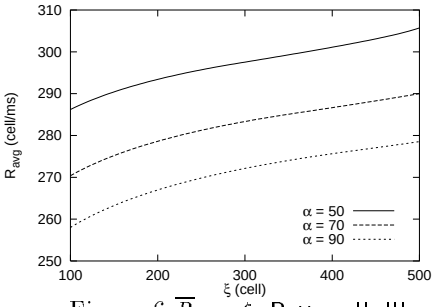


Figure 6 $\overline{R}$ vs. ξ: Pattern II, III

average throughput. In Figure 6, $\overline{R}$ is found to monotonically increase as ξ increases, but for a given ξ, $\overline{R}$ decreases as α grows.

Performance Comparison among Three Control Patterns: In Figure 7, the normalized $\overline{R}$'s are plotted against α with different values of ξ, corresponding to different control patterns. We have made the following observations. For any given α, the equilibrium state governed by Pattern I represents the optimal equilibrium state in terms of average throughput, queuing delay, and delay variation. Thus, we define control Pattern I as the optimal control pattern/state. For a given ξ, $\overline{R}$ monotonically decreases as α increases for all three patterns. $\overline{R}$ of Pattern II and III with a smaller ξ decays faster as α increases. For any given α, increasing ξ can improve $\overline{R}$, but when $\xi \geq Q_{max}$, $\overline{R}$ cannot be improved any further by increasing ξ. So, the average throughput $\overline{R}$ is upper bounded by curve $\xi \geq Q_{max}$, thus providing information on optimal buffer allocation to a VC for different α's. The larger α, the more sensitive to ξ the average throughput $\overline{R}$ is. In general, a smaller α leads to better equilibrium-state performance.

Since Q_{max} is proportional to α, we can adjust α to an appropriate smaller value such that $Q(t)$'s fluctuation is bounded by ξ and then the system operates in the optimal equilibrium state (under control Pattern I). But α should not be too small since a small α degrades transient-state performance.

5.3 Transient-State Rate Control and Performance Analysis

Transient State and Its Rate-Control Algorithm: The transient state is defined as a state between any initial state and an optimal equilibrium state. The goal of our control algorithm is to drive the system from any initial state into the optimal equilibrium state as quickly as possible while maintaining a high throughput. Since rate increase or decrease can only make $R(t)$ fluctuate around the designated bandwidth, but cannot adjust the rate-fluctuation amplitude that determines Q_{max}, we need a higher-order rate control which directly adjusts the rate parameter α (i.e., $\frac{dR(t)}{dt}$) (β is also adjusted by setting $\alpha\Delta/(1-\beta) = \mu$) to reduce the rate-oscillation amplitude. There are other reasons necessitating the dynamic adjustment of α. In a real network, the round-trip delay τ varies with time. Thus keeping Q_{max} at a given level requires α to vary with time. In this paper, however, we only consider how to reduce α to ensure Q_{max} does not grow beyond ξ while achieving a good transient-state response.

Let α_0 be the initial source rate-increase parameter. Application of the α-control rule n times will yield a sequence $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$. For a good transient response, we use an exponential control rule which is defined by

$$\alpha_n = e^{-\lambda n}\alpha_0 \quad (\lambda > 0). \tag{5.19}$$

where λ specifies the speed of reducing α.

Note that α should not be reduced further as long as $Q_{max}(\alpha_n) < \xi$, where Q_{max} is a function of α as shown in Eq. (5.3), since too small a value of α will slow down the transient system behavior, or even disable the capability of grabbing the spare bandwidth created by other idle VCs. So, the source should stop execution of the α-control rule as soon as α_n reaches its optimal value α^*:

$$\alpha^* \stackrel{\triangle}{=} \max_{i \in \{1,2,3,\ldots\}} \{\alpha_i \mid Q_{max}^{(i)} < \xi\}, \tag{5.20}$$

where $Q_{max}^{(i)}$ is the maximum queue length for $\alpha_i = e^{-\lambda}\alpha_{i-1}$, $\alpha_0 > \alpha^*$.

Analytical Solutions and Performance Analysis for Transient State: We assume $\alpha_0 \geq \alpha^*$, and focus on the first-cycle dynamic behavior with initial rate $R_0 > \mu$. Transient state flow-control also divides control into three patterns, which are defined similarly to those for equilibrium state. Since for $R_0 > \mu$ the system typically operates in transient state under Patterns II and III, our analysis here will focus on these two patterns. Notice that for these two patterns, when $Q_{max} > \xi$, $R(t)$ restarts rate-increase from R_c with a smaller increase rate of $\alpha e^{-\lambda}$ instead of α.

The detailed descriptions of control patterns and derivations of their corresponding analytical expressions are available in [12]. Here we only present some numerical results on the transient-state performance. The network condition remains the same as in Section 5.2. But we use $R_0 = 4\mu$ and $\lambda = log\ 2$[1] here. In Figure 8–9, we observe that for a given ξ a larger α not only results in a higher transient-state average throughput, but also a shorter transient-cycle length. Notice that this observation is the opposite of what we observed in the equilibrium state where a small α leads to a high throughput. These observations suggest that our SES algorithm start sending data with a larger initial rate-control parameter α_0, but make α smaller as system

[1] This implies $\alpha_i = \frac{1}{2}\alpha_{i-1}$, just a left-shift operation which is easy to implement. But λ can take any other positive number.

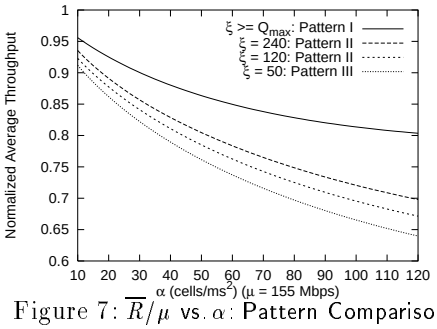


Figure 7: $\overline{R}/\mu$ vs. α: Pattern Comparison

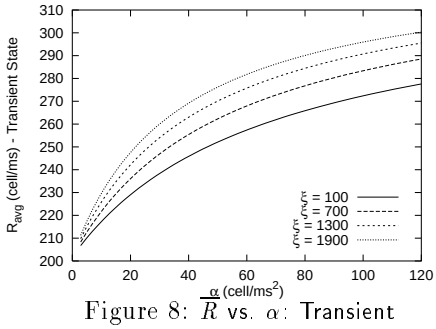


Figure 8: $\overline{R}$ vs. α: Transient

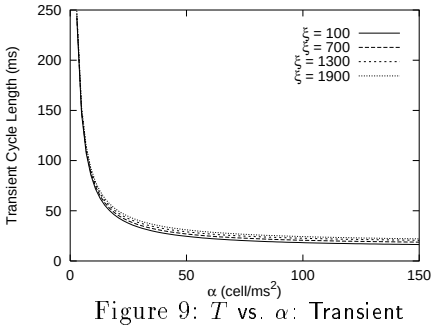


Figure 9: T vs. α: Transient

converges to the optimal equilibrium state. This observation is consistent with the conclusion: "a sharp rate reduction in the transient state and a smaller rate reduction during the equilibrium state" [2].

6 Dynamics of Multiple ABR Connections

Consider a scenario where N flow-controlled VCs share a common bottleneck. The parameters describing the VC$_i$ in this multi-connection system are given below.

ξ_i: $\triangleq \frac{MCR_i}{\sum_{j=1}^{J} MCR_j}\xi$ is buffer share for VC$_i$

μ_i: $\triangleq \frac{MCR_i}{\sum_{j=1}^{N} MCR_j}\mu$ is bandwidth share for VC$_i$

MCR_i: MCR for VC$_i$; we assume $MCR_i > 0$

$Q_i(t)$: Queue length at the bottleneck node for VC$_i$

$R_i(t)$: Source rate for VC$_i$

$\alpha^{(i)}$: Rate increase parameter for VC$_i$

$T_b^{(i)}$: Backward feedback delay for VC$_i$

$T_f^{(i)}$: Forward delay for VC$_i$

Δ_i: Time interval of rate-update for VC$_i$

At the bottleneck switch, the total buffer capacity ξ is statically allocated to N existing VCs, each with a buffer share proportional to its MCR. J ($\leq N$) active VCs dynamically share the bottleneck link bandwidth μ, each VC being served in a rate proportional to its MCR. To make the analysis tractable, we ignore the scheduling time at the switch, and also consider the assigned bandwidth share as the target bandwidth share (instead of the realized bandwidth), which slightly under-estimates the throughput, but still reflects the system dynamic behavior. Then, all the expressions derived in Section 5 can be applied to the multiple-connection case with the target bandwidth and buffer capacity substituted by their shares. Next, we present two examples.

Example 1: Buffer Requirement and Average Throughput. We consider a case where there are $N = 4$ identical ABR VCs with: $R_0^{(i)} = 183.5$ cells/ms, $MCR_i = 18.35$ cells/ms, $T_b^{(i)} = T_f^{(i)} = 1$ ms, $\Delta_i = 1$ ms, $Q_h^{(i)} = 50$ cells, $Q_l^{(i)} = 25$ cells, and $\alpha_0^{(i)} = 11.45$ cells/ms^2. Assume that 4 VCs start sending cells at the same time over the bottleneck link with $\mu = 367$ cells/ms (155 Mbps).

For the rate scheme, the 4 VCs share a common FIFO output queue $Q(t)$ at the bottleneck link. Using the equations derived for Pattern I (describing rate scheme), we obtain the evolutions of $R(t) = 4R_i(t)$ and $Q(t)$, as shown in Figure 10. In transient state, a large queue build-up, $Q_{peak} = 2207.5$ cells, is observed. But in equilibrium state, Q_{max} is just 493 cells, about 1/5 of Q_{peak}. For lossless transmission, a buffer size larger than 2207 cells is required to prevent cell-loss during the short transient duration even though only 22% of buffer space will be utilized during the long equilibrium duration. The resulting equilibrium-state average throughput is 319.35 cells/ms (or $\overline{R}/\mu = 0.87$).

With the proposed scheme, we assume the bottleneck switch's $\xi = 500$ cells. Then, each VC's $\xi_i = 125$ cells and $\mu_i = 91.75$ cells/ms since all VCs are identical. The 4 VCs each have their own output queue at the bottleneck switch. Using the equations derived for Pattern I, II, and III which characterize the proposed scheme, we compute the evolutions of $R_i(t)$ and $Q_i(t)$ for both transient and equilibrium states. Since the 4 VCs are identical, we have $Q(t) = 4Q_i(t)$ and $R(t) = 4R_i(t)$. As shown in Figure 10, $R(t)$ experiences just one cycle of transient state with $\alpha_0^{(i)} = 11.45$ cells/ms^2 and then enters the equilibrium with $\alpha_1^{(i)} = 5.725$ cells/ms^2 ($\lambda = log\ 2$). In the transient state, $Q(t)$ is bounded by buffer size $\xi = 500$ without any cell-loss due to buffer overflow, and $Q_{max} = 356$ cells in equilibrium state. The resulting equilibrium-state average throughput is 336.7 cells/ms (i.e., $\overline{R}/\mu = 0.92$), which is higher than that of the rate-based scheme.

This example shows that the proposed scheme requires a much smaller (nearly 5 times less) buffer size to guarantee lossless transmissions and achieves higher average throughput than the rate-based scheme.

Example 2: Bandwidth Guarantees and Fairness. Here, we consider two ABR connections with different parameters. For connections: $MCR_1 =$ 12 cells/ms, $\alpha_0^{(1)} = 22.9$ cells/ms^2, $R_0^{(1)}$=0 cells/ms; $MCR_2 = 24$ cells/ms, $\alpha_0^{(2)} = 45.8$ cells/ms^2, $R_0^{(2)}$=0 cells/ms. For networks: $\mu = 367$ cells/ms, $\xi = 450$ cells, $Q_h = 45$ cells, $Q_l = 22.5$ cells, $T_b^{(1)} = T_b^{(2)} = 1$ ms, $T_f^{(1)} = T_f^{(2)} = 1$ ms, $\Delta_1 = \Delta_2 = 1$ ms. Then, $\mu_1 = \mu/3 = 122.3$ cells/ms, $\mu_2 = 2\mu/3 = 244.7$ cells/ms, $Q_h^{(1)} = Q_h/3 = 15$ cells, $Q_h^{(2)} = 2Q_h/3 = 30$ cells, $Q_l^{(1)} = Q_l/3 = 7.5$ cells, $Q_l^{(2)} = 2Q_l/3 = 15$ cells,

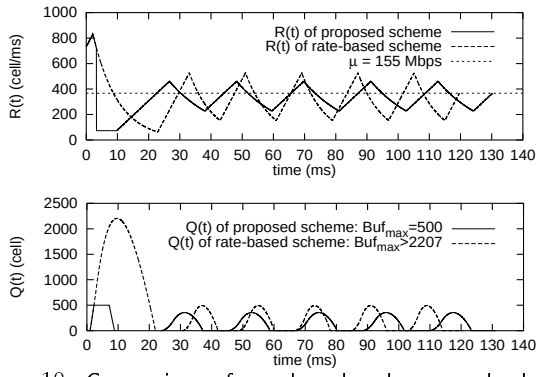


Figure 10: Comparison of rate-based and proposed schemes.

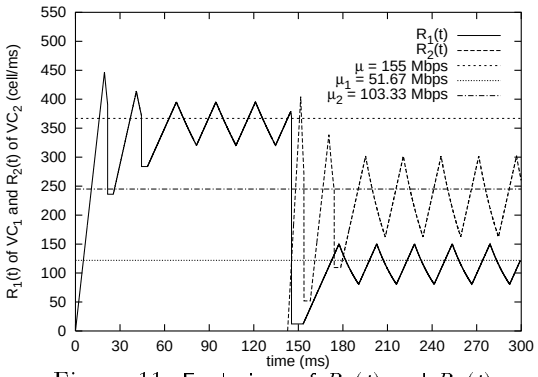


Figure 11: Evolutions of $R_1(t)$ and $R_2(t)$.

$\xi_1 = \xi/3 = 150$ cells, $\xi_2 = 2\xi/3 = 300$ cells.

We assume that VC_1 starts sending data at $t = 0$ and VC_2 starts sending data at $t = 142.78$ ms when VC_1 has already reached the optimal equilibrium state. For the proposed scheme,we compute the evolutions for $R_1(t)$ and $R_2(t)$ for transient and equilibrium states as shown in Figure 11. We observe that after 2 transient cycles (46.95 ms) $R_1(t)$ converges to μ, instead of its share μ_1. This is because there are no other VCs sharing μ with VC_1, and thus, VC_1 grabs all available bandwidth μ (ABR). At $t = 142.78$ ms, VC_2 starts sending data cells, then the scheduler at the switch assigns μ_1 to VC_1 and μ_2 to VC_2 as their target bandwidth shares. VC_2 starts competing for bandwidth in its transient cycles. In the mean time, $R_1(t)$ starts to give up the bandwidth beyond its share μ_1. Note that VC_1's α remains the same since it has reached its optimal value 5.725 cells/ms^2. After 2 transient cycles (47.52 ms), both $R_1(t)$ and $R_2(t)$ converge to their shares μ_1 and μ_2. Note that by properly reducing $\alpha^{(1)}$ and $\alpha^{(2)}$, not only do $R_1(t)$ and $R_2(t)$ converge to their shares, but also $Q_1(t)$ and $Q_2(t)$ are confined to the regimes bounded by ξ_1 and ξ_2, since the resulting $Q^{(1)}_{max} = 131$ and $Q^{(2)}_{max} = 263$, respectively.

This example shows that the proposed scheme can provide a bandwidth guarantee to each VC and achieve a fair bandwidth share among competing connections according to their MCRs. As previously discussed, a bandwidth guarantee is hard to achieve by the credit scheme, as it does not explicitly control transmission rate. These two examples also show that under the proposed scheme $R(t)$ and $Q(t)$ can rapidly converge to the optimal operating regime (within two cycles of the transient state).

7 Conclusions

In this paper, we proposed and evaluated an integrated credit- and rate-based flow-control scheme. The proposed scheme combines the merits and overcomes the weakness of the two schemes by exercising direct control over both bandwidth and buffer resources. Unlike the previous flow-control schemes and analyses, we included the buffer capacity as an important constraint in the design and analysis of the proposed scheme. From the analyses, we identified the optimal control pattern and developed a 2-dimensional rate-control scheme to drive the system to the optimal control pattern. Through examples, it is shown that our scheme outperforms the other existing schemes in terms of buffer requirement (with lossless transmission), average throughput, bandwidth guarantees, fairness, and network utilization. The simulation results have verified the analytical results for the single-connection case. We are currently extending the simulator to the multiple-connection case.

References

[1] H. T. Kung, T. Blackwell, and A. Chapman, "Credit update protocol for flow-controlled ATM networks: statistical multiplexing and adaptive credit allocation," in *Proc. of ACM SIGCOMM*, pp. 101–115, 1994.

[2] N. Yin and M. G. Hluchyj, "On closed-loop rate control for ATM cell relay networks," in *Proc. of IEEE INFOCOM*, pp. 99–109, June 1994.

[3] S. Sathaye, *ATM Forum traffic management specifications Version 4.0*, ATM Forum contribution 95-0013R7.1, August 1995.

[4] *IEEE Network Magazine-Special Issue on ATM Flow Control: Rate vs. Credit*, volume 9, March/April 1995.

[5] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd edition, 1992.

[6] N. Yin, "Analysis of a rate-based traffic management mechanism for ABR service," in *Proc. of GLOBECOM*, pp. 1076–1082, November 1995.

[7] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Analysis of rate-based congestion control for ATM networks," *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 60–72, April 1995.

[8] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Analysis of rate-based congestion control algorithms for ATM networks—Part 1: steady state analysis—," in *Proc. of GLOBECOM*, pp. 296–303, November 1995.

[9] M. Ritter, "Network buffer requirements of the rate-based control mechanism for ABR services," in *Proc. of IEEE INFOCOM*, pp. 1190–1197, March 1996.

[10] J. Bolot and A. Shankar, "Dynamical behavior of rate-based flow control mechanism," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 35–49, April 1990.

[11] F. Bonomi, D. Mitra, and J. Seery, "Adaptive algorithms for feedback-based flow control in high-speed, wide-area ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1267–1283, September 1995.

[12] X. Zhang and K. G. Shin, "On integrated rate and credit flow control," in preparation.