Integrated Rate and Credit Based Flow Control for Unicast 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
313-763-0391 (voice); 313-763-4617 (fax)
E-mail: {xizhang,kgshin}@eecs.umich.edu

Qin Zheng

Gigapacket Networks, Inc.
25 Porter Road
Littleton, MA 01460

E-mail: zheng@gigapacket.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 by maximizing average throughput and bounding end-to-end delay. By applying the second-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. Using hop-by-hop credit flow control, the proposed scheme ensures lossless transmission with a finite buffer capacity while maintaining high network utilization. The source dynamically adjusts its transmission rate and rate-control parameters according to both the network congestion status and whether or not the system states are in the target operating regime.

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 the equilibrium states as functions of rate-control parameters, feedback delay, link bandwidth, and buffer capacity. Based on the analytical results, we identify the optimal equilibrium state and propose the second-order rate control algorithm for transient-state flow control. We derive a sufficient condition which guarantees the second-order rate control to drive the system from any transient state to an optimal equilibrium state. The worst-case performance bounds, derived as the closed-form functions of flow-control parameters, provide an analytical means of evaluating the performance of the second-order rate control in terms of convergence speed and buffer utilization. The analytical results for both single and multiple connection scenarios have shown the proposed scheme to be stable and efficient in that the source rate and bottleneck queue length rapidly converge to a small neighborhood of the designated operating point. Also, presented are examples showing that the proposed scheme outperforms the other existing schemes.

Index Terms — ATM networks, ABR service, feedback flow control, credit-based flow control, rate-based flow control, lossless transmission, shared and distributed resource control.

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 N00014-94-0029. 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.

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 Virtual Circuit – VC) from its source to destination. A connection runs through a sequence 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, 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 sort of 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. Each 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* [1–4] and *rate* [5–9] schemes have received most attention [10]. 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 [11]. 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, and the initial rate [12–15]. 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 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 shares guaranteed among competing connections,
- Maximum network utilization.

Using the first-order fluid approximation [16–18], we model the proposed scheme and analyze the system's dynamic behavior for ABR service under the most stringent traffic condition. In previous

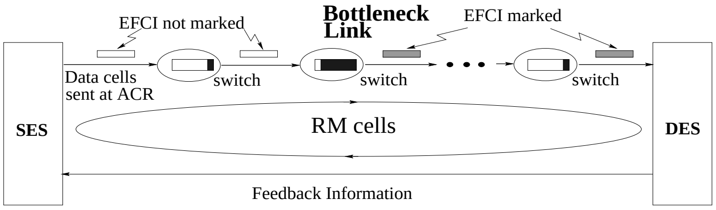


Figure 1: Basic configuration of the rate-based flow control scheme.

performance analyses, the maximum queue length Q_{max} was treated as a free parameter under the unrealistic assumption of infinite buffer capacity [5, 12–15, 18–20]. 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 for queue build-ups and average throughput in both transient and equilibrium states. Also, we derive a set of expressions for computing the evolutions of rate and queue length functions for both transient and equilibrium states. From the analysis, we identify the optimal rate-control pattern and conclude that just exercising increase/decrease rate control is not effective enough to have the system converge to the optimal operating regime specified by the allocated bandwidth and buffer capacity. To ensure convergence from any initial state to the optimal equilibrium state, we develop a second-order rate-control algorithm. A sufficient condition is derived to show the feasibility of the second-order rate control. We also derive the worst-case performance bounds, as the closed-form functions of flow-control parameters, which not only provide performance measures for the second-order rate control, but also offer the insight on how to select appropriate parameters for the second-order rate control. Applying the second-order rate control in the transient state shows that the system rapidly converges to the designated optimal operating regime.

To support per-VC queueing and fair bandwidth sharing, we apply a weighted round-robin (WRR) scheduling to transmit each connection's cells at a frequency proportional to its MCR (Minimum Cell Rate). Using two multiple-connection examples, it is shown that the proposed scheme outperforms the other existing schemes in terms of buffer requirement, lossless transmission, bandwidth share guarantees, and average throughput, and makes system states converge to the optimal operating regime quickly.

This paper is organized as follows. In Section 2, we describe and compare 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. In Section 6, we propose the second-order rate control algorithm for the transient state flow control, and describe its properties and transient-state performance analysis. Section 7 analyzes the proposed scheme's performance for the multiple-connection case through examples. The paper concludes with Section 8.

2 Rate vs. Credit, and Why Interworking ?

We first describe rate and credit schemes, and then compare them in terms of structures and performance, arguing for the need to integrate them.

2.1 The Rate Scheme

We briefly describe the operation mode of the rate-based congestion-control mechanism, drawing mainly from the ATM Forum traffic management standard [9]. The standard only specifies the behaviors of the

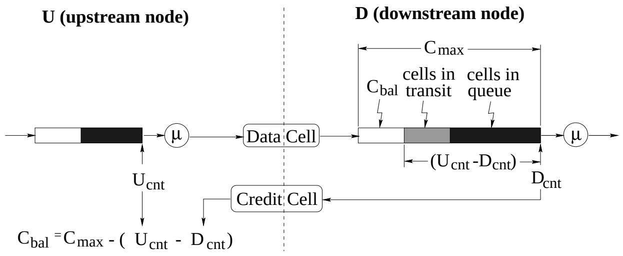


Figure 2: Credit-based flow control.

Source End System (SES) and the Destination End System (DES), leaving the implementation of ATM switches to manufacturers. Figure 1 illustrates the basic configuration of the rate scheme. SES is allowed to transmit cells at a rate up to the Allowed Cell Rate (ACR) which varies dynamically, depending on network congestion status. The ABR client negotiates with the network on the Initial Cell Rate (ICR), the Minimum Cell Rate (MCR), and the Peak Cell Rate (PCR) at the time of connection setup. ACR is varied between MCR and PCR.

SES sends a forward Resource Management (RM) cell every N_{rm} data cells (or periodically) to collect congestion information. Upon arrival of a forward RM cell at DES, the cell is returned as a backward RM cell to SES. The current congestion information on network resources is provided by DES and intermediate switches, which may modify the content of the backward RM cell. When the backward RM cell arrives at SES, the SES adjusts its ACR according to the Congestion Indication (CI) bit and the Explicit Rate (ER) field in the RM cell. If SES receives a backward RM cell with CI=1, it decreases ACR *multiplicatively* (but not below MCR); otherwise, it increases ACR *additively* (but below PCR). When a switch/DES experiences a severe congestion, it can reduce the ER field of RM cell, requesting that the source quickly reduce its transmission rate to the switch's acceptable level. SES sets its ACR to ER if it is lower than the current ACR.

The CI bit and the ER field in an RM cell are set according to a switch's data queue length. There are two types of switch architectures, which involve different flow-control actions.

***Explicit Forward Congestion Indication (EFCI) based switch*:** If the queue length grows beyond an upper threshold Q_h, the switch marks the EFCI bit in the header of passing-by data cells to indicate congestion (see Figure 1), until the queue length drops below a lower threshold Q_l. DES sets the CI bit in each backward RM cell according to EFCI status of the most recent data cell received.

***Explicit Rate (ER) based switch*:** ER-based switches are equipped with an *intelligent marking* and explicit rate setting capability. This enables switch to selectively reduce the rates of ABR sources by marking the CI bit, or setting the ER field, of the corresponding VCs with larger ACRs according to the degree of congestion in forward and/or backward RM cells.

Both types of switches may coexist in a single network environment and also backward RM cells may be generated by a switch to achieve quick congestion notification/dissipation. A detailed description of rate-based flow control can be found in [9].

2.2 The Credit Scheme

The operation of the credit-based flow control is shown in Figure 2 for a single-hop VC. At the upstream node U, every VC keeps a credit balance C_{bal} which is initialized to C_{max}, the maximum number of buffer slots allocated at the downstream node for this VC. Each time U sends a data cell on the VC, this VC's

C_{bal} is decreased by one. As long as the C_{bal} is positive, the upstream node U can transmit cells on that VC. For each VC, the upstream node U keeps a running count U_{cnt} of all the data cells it has transmitted, and the downstream node D keeps a running count D_{cnt} of all data cells it has forwarded. D encloses the up-to-date value of D_{cnt} in the credit record field of the credit cell, which is transmitted to U periodically or once every certain number of data cells. Upon receiving the credit cell with D_{cnt}, U updates its credit balance, $C_{bal} = C_{max} - (U_{cnt} - D_{cnt})$, for that VC (see Figure 2). Note that the non-negative quantity $U_{cnt} - D_{cnt}$ represents the outstanding cells/credits which correspond to those cells of the VC U has transmitted but D has not yet forwarded. These are "cells in transit" plus "cells in queue" when D sends the credit cell, containing D_{cnt}, to node U. The frequency of sending feedback credit cells can be controlled periodically or one credit cell per N data cells forwarded by the downstream node D for some positive integer N. Periodically sending credit cells (thus based on time) can help system recover from possible loss of credit cells, but may introduce more overhead traffic. Sending a credit cell per N data cells is more economical since it keeps overhead traffic at a fixed percentage of data traffic. However, a credit-check cell [3] needs to be sent periodically to ensure the recovery from possible loss of credit cells.

There are two important features for the credit scheme described above. First, the scheme ensures strict lossless transmissions since C_{bal} is always smaller than, or equal to, currently available buffer space at the downstream node and never exceeds C_{max}. Second, if C_{max} is greater than the product of the bandwidth capacity, μ, and the link round-trip delay, τ_l, between U and D, then the VC can run at the full link speed as long as all other VCs are idle. Note that τ_l represents the sum of propagation delay, processing delay, and credit cell transmission interval. In the credit scheme, the average transmission rate during τ_l can be controlled *indirectly* by C_{max} and the frequency of credit cells.

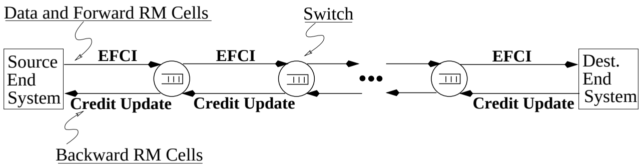
2.3 Comparison of Rate and Credit Schemes and Their Problems

The rate scheme regulates a connection's bandwidth by directly controlling its source cell-transmission rate according to network congestion information. Using RM cells and EFCI 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. To maintain high utilization, the credit scheme always attempts to use all available buffers. The goal of credit scheme is to ensure lossless transmission with a given finite buffer capacity while maintaining high bandwidth 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: It is difficult to design buffers with the rate scheme. As was analyzed in [12–15], the buffer requirement for the rate scheme is very large and increases with feedback delay, the number of active connections, and the initial rate. However, the exact values of the network delay of each connection and the number of active connections over a given link are usually unknown *a priori*, which makes it very hard to predict the required buffer space. 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 band-width-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.



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

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. In the rate scheme, the feedback information on available bandwidth of a bottleneck link located far away can take a significant amount of time to reach the source node, thus making it difficult to achieve high utilization of bandwidth. Moreover, if lossless transmission or a low loss rate 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 apply *direct* control over *both* bandwidth and buffer resources.

3 The Proposed Scheme

Based on 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 3. 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 queue length threshold; (2) *buffer congestion*, if credit balance $C_{bal} = 0$. If a buffer congestion occurs at a switch, the switch generates a backward RM cell (with BN=1) and sends it back to the source for a quick notification/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 to an appropriate smaller value R_c, which is less than the bottleneck bandwidth μ, but larger than its MCR. How to calculate R_c will be discussed later (see Eq. (5.11));

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

Table 1: A pseudocode for Source End System (SES).

- exponentially reduce the rate-increase parameter which is the second-order rate control and will be discussed later (see Eq. (6.1)).

These enhanced features in structures and algorithms enable the proposed scheme to cope with the following practical problems that the other two schemes cannot deal with. For given buffer capacity (always finite and usually specified by the vendor), 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, PCR, ICR, 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 node, the destination node, and all switch nodes between the source and destination nodes.

3.2.1 The Source Node Algorithm

The source control algorithm is described by the pseudocode in Table 1. This algorithm deals mainly with two events: sending data cells (lines: 03–10 of the pseudocode) and receiving backward RM cells (lines: 11–25 of the pseudocode). When the rate-control timer expires, it must first check if credit balance C_{bal} is positive. If $C_{bal} > 0$, it sends a data cell to the downstream node and then increases the count and decreases the credit balance by one (book-keeping). The source sends a forward RM cell once every N_{rm} data cells. Finally, the rate-control timer is reset to the next cell time according to the current

00. Local Variables: U_{cnt}, C_{bal}, $Data\_que$, $Data\_que\_len$, $Local\_VC\_CCR$; $Local\_VC\_ER$
01. $Local\_VC\_CI := 0$; ! Local VC congestion indicator
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** ($C_{bal} = 0$) send RM($DIR := backward, CU := U_{cnt}, CI := 1, ER := \mu, BN := 1$)
09. **if** ($Data\_que\_len \geq Q_h$) $Local\_VC\_CI := 1$;
10. **if** (receive output link ready signal and $Data\_que\_len > 0$) ! Receiving link_ready signal event
11. schedule all active VCs {
12. **if** ($C_{bal} > 0$)
13. remove Data cell from head of $Data\_que$;
14. **if** ($Data\_que\_len \leq Q_l$) $Local\_VC\_CI := 0$;
15. forward Data($EFCI := EFCI \oplus Local\_VC\_CI$) cell;
16. $U_{cnt} := U_{cnt} + 1$; $C_{bal} := C_{bal} - 1$; ! Book-keeping
17. **else**
18. send RM($DIR := backward, CU := U_{cnt}, CI := 1, ER := \mu, BN := 1$) cell }
19. **if** (receive RM(DIR, CI, CCR, ER, CU, BN) cell) ! Receiving an RM cell event
20. **if** ($DIR = forward$)
21. $Local\_VC\_CCR := CCR$; $Local\_VC\_ER := ER$;
22. send RM($DIR = forward$) cell;
23. **else**
24. $C_{bal} := C_{max} - (U_{cnt} - CU)$; ! $CU = D_{cnt}$
25. send RM($DIR = backward, CU := U_{cnt}, CI := CI \oplus Local\_VC\_CI$) cell;

Table 2: The pseudocode for Intermediate Switch System (ISS).

value of ACR. Upon receiving a backward RM cell from a VC, first the source updates its local credit balance using the CU value of the RM cell and then proceeds to rate control. In rate control, the source must first check if this VC is already in buffer congestion state. If the VC was not in buffer congestion state, then (1) if $BN = 0$ then the source *additively* increases its ACR or *multiplicatively* decreases its ACR depending on the CI bit (set ACR equal to MCR if C_{bal}=0); (2) If $BN = 1$ then the source enters buffer congestion state, and exercises the buffer congetion control by setting ACR to the cut-down rate R_c (see Eq. (5.11), exponentially reducing rate-increase parameter AIR (see Eq. (4.1), and Eq. (6.1) with $\lambda = log\ 2$), and accordingly adjusting rate decrease parameter MDF (see Eq. (4.1), and $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$ in Section 5.2). If this VC is already in the buffer congestion state, then the source ACR stays with the cut-down rate 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 (Our scheme allows either per-N_{rm} data cells or periodic rate-update control. For simplicity only the per-N_{rm} data cells scheme is presented in the pseudocode.)

3.2.2 The Switch Node Algorithm

The pseudocode of the switch algorithm is given in Table 2. There are three main events in the switch algorithm.

Receiving data: (lines: 02–09 of the pseudocode) forward the data cell if the output link is ready and if $C_{bal} > 0$; enqueue the data otherwise. Mark the 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.

Receiving link-ready signal: (lines: 10–18 of the pseudocode) schedule the active VCs using WRR algorithm. Dequeue a data cell for the scheduled VC. If queue length drops below Q_l, the local

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($DIR = forward, CI, CCR, ER, CU, BN$) cell) ! Receiving RM cell event
06. send RM($DIR = backward, CU := U_{cnt}, CI := CI \oplus Local\_VC\_CI, BN := 0$) cell;

Table 3: The pseudocode for Destination End System (DES).

congestion indication bit ($Local\_VC\_CI$) is unmarked. Generate and send an RM cell directly back to the source with BN=1, $CI = 1$, and ER = bandwidth if $C_{bal} = 0$ after credit bookkeeping.

Receiving an RM cell: (lines: 19–25 of the pseudocode) if it is a forward RM cell, record contents of the forward RM cell and forward it to the downstream node; if it is a backward RM cell then update the local credit-balance using the RM cell's CU field, fill in the local count and congetion status in the RM cell, and then send the RM cell to the upstream node with the updated credit and congestion information.

3.2.3 The Destination Node Algorithm

There are two events to consider in the destination algorithm: receving data cells (lines: 01–04 of the pseudocode) and receiving RM cells (lines: 05–06 of the pseudocode). 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. Its pseudocode is given in Table 3.

3.3 Implementation Issues

While possessing the merits of *both* rate and credit schemes, the extra implementation cost of the integrated scheme is minimum, especially if using switches which already support per-VC queuings, e.g., the FORE ASX-200E and DEC AN2 switches. The complexity of the proposed scheme is basically the combination of complexity for the EFCI-based rate scheme and the credit scheme. Although R_c needs to be computed based on the estimated bandwidth sent back from the bottleneck switch (see Eq. (5.11)), the cost of the rate-control part still remains at the same cost level of EFCI-based rate scheme, because (1) R_c needs to be computed only when buffer congestion occurs, (2) there is no need to compute and keep track of the explicit rate at each switch all the time, and (3) no explicit rate control is invoked at the source. This is also the reason why we will, in Section 7, compare the proposed scheme with the EFCI-based rate scheme, instead of ER-based rate scheme. Moreover, the buffer congestion indication needs only one bit, which is taken care of by BN bit of the standardized ATM Forum RM-cell specification and thus does not incur any extra cost. On the other hand, since buffer capacity has been treated as a control parameter in the integrated scheme, the maximum buffer memory requirement is significantly reduced compared to the rate scheme which needs a large buffer space to reduce loss rate. The computation of credit balance involves only add/subtract operations which can be easily implemented by simple hardware. When buffer congestion occurs, R_c and a new α can be computed by quantization and look-up tables, instead of more complicated multiply/divide operations. For sending credit-update information to the upstream nodes, only two extra bytes are added into the RM-cell format such that the extra bandwidth required for transmitting the signaling information is minimized.

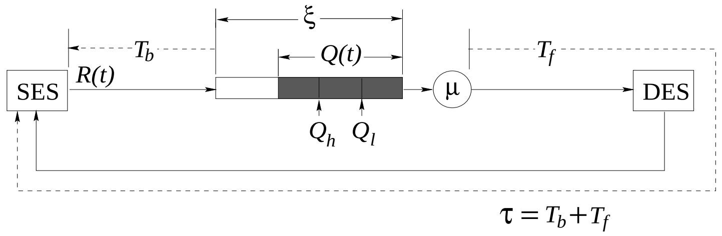


Figure 4: The system model for a virtual circuit.

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, which characterizes the system with coupled time-delayed differential equations [17, 18]. In this model, we use the real-valued deterministic functions $R(t)$ and $Q(t)$ to approximate a discrete-valued stochastic rate process $\mathcal{R}(t)$ at the source and a queue length process $\mathcal{Q}(t)$ at the bottleneck node, respectively. 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 [5, 12–15, 18–20].

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 [5, 12–15, 18–20]. 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 show that by properly adjusting the flow-control parameters, the proposed scheme can guarantee lossless transmissions while achieving high average throughput for a given finite buffer capacity. 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 does represent such applications as file transfer and image retrieval. Moreover, this source model allows us to examine the proposed scheme under the most stressful condition. Figure 4 depicts the system model for each VC flow-controlled by the proposed scheme.

4.1 System Description

The system with the proposed flow-control scheme is characterized by the following parameters (see Figure 4):

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

The backward delay, T_b, is defined as the duration from the time when a congestion signal is generated and sent back from the bottleneck node to the time when it is received by the source, including propagation and processing delays. As mentioned earlier, there are two types of congestion in our model: *link bandwidth* and *buffer capacity* congestions. The bandwidth congestion occurs when $Q(t) \geq Q_h$, for which our algorithm sends data cells with the EFCI bit marked. The buffer congestion occurs when $C_{bal} = 0$, which represents a much more severe congestion condition. So, upon occurrence of a buffer congestion, our algorithm generates and sends a backward RM cell with BN = 1 (Figure 3) requesting the source to reduce its transmission rate. This RM cell experiences a delay of T_b to reach the source node. Note that T_b also represents the delay from the instant the rate-control information is sent by the source to the time when the rate control starts acting on the bottleneck.

The forward delay, T_f, is measured from the detection of bandwidth congestion at the bottleneck to the time when the congestion-indication feedback reaches the source via the destination node. Thus, $\tau = T_b + T_f$ represents the VC's round-trip delay. We use the synchronous model for rate control in which the fixed (periodic) rate-update interval Δ is usually a fraction of τ. During the establishment of an ABR connection, the client specifies to the network both its PCR and MCR for the requested connection. We explicitly take MCR into modeling since it is closely related to the minimum bandwidth guarantee for an ABR connection. μ and ξ are the bottleneck's bandwidth and buffer capacity, respectively, which represent two types of network resource constraints. Q_h and Q_l are used, respectively, for traffic overload and underload detections. Based on the proposed control algorithms in Section 3.2, the additive increase and the multiplicative decrease of rate during the n-th rate-update interval are expressed as:

$$R_n = \begin{cases} R_{n-1} + a; & \text{Additive increase} \\ bR_{n-1}; & \text{Multiplicative decrease} \end{cases} \tag{4.1}$$

where a (AIR in the source node algorithm) is the rate increment and b (MDF in the source node algorithm) is the rate decrease factor. Thus, the rate adjustment at SES can be modeled by "linear-increase" and "exponential-decrease" in a continuous domain with the following expressions [5]:

$$R(t) = \begin{cases} R(t_0) + \alpha(t - t_0); & \text{Linear increase} \\ R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}}; & \text{Exponential decrease} \end{cases} \tag{4.2}$$

where t is the current time; t_0 the time of the last rate-update; $\alpha = a/\Delta$ and $\beta = 1 + log\ b$ for a rate-adjustment interval Δ.

4.2 System State Equations

The system state is specified by two state variables: $R(t)$ (source rate function) and $Q(t)$ (bottleneck's queue-length function). According to the proposed control algorithms, the system state equations for a VC containing a single bottleneck are given by the following three sets of equations, depending on whether rate or credit control is in operation.

Phase 1: rate-control increase if $Q(t - T_b) < Q_h$,

$$R(t) = R(t_0) + \alpha(t - t_0) \qquad \text{and} \qquad Q(t) = \int_{t_0}^{t} [R(v - T_b) - \mu]dv + Q(t_0) \tag{4.3}$$

Phase 2: rate-control decrease if $Q(t - T_b) \geq Q_h$,

$$R(t) = R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}} \qquad \text{and} \qquad Q(t) = \int_{t_0}^{t} [R(v - T_b) - \mu]dv + Q(t_0) \tag{4.4}$$

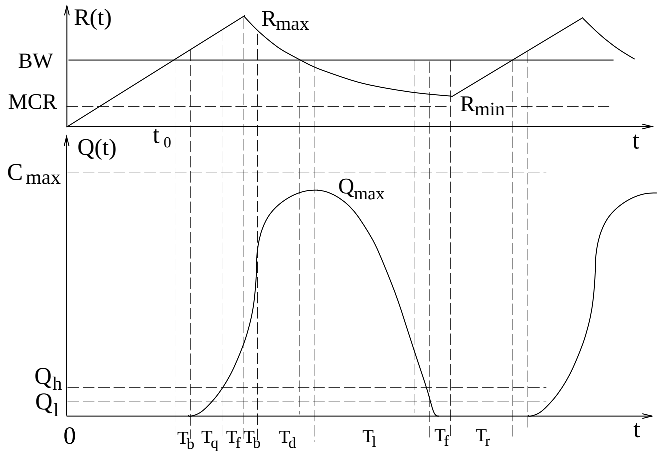


Figure 5: Dynamic behavior of $R(t)$ and $Q(t)$ for control pattern I.

Phase 3: credit-control if the source receives a feedback of $C_{bal} = 0$ (buffer congestion),

$$R(t) = R_c, \quad (R_c \geq MCR) \qquad \text{and} \qquad Q(t) = \xi - (\mu - R_c)(t - T_b - t_0). \tag{4.5}$$

Here $R(t)$ represents the fluid approximation to the throughput of cell transmissions. The average throughput is then given by $\lim_{t\to\infty} \frac{1}{t}\int_0^t R(v)dv$. $Q(t)$ is an approximation to the queue-length process at the bottleneck node. R_c is the cut-down rate set by the source when it receives a (BN=1) RM cell (notifying $C_{bal} = 0$) from the bottleneck node. The non-linear dynamics of the rate functions reflect the fact that $Q(t) \in [0, \xi]$.

This first-order fluid model has been shown in [17] to be a good approximation when the system is heavily-loaded. The persistent source model accommodates an accurate approximation since it represents a heavy traffic condition.

5 Analysis of a Single ABR Connection

In this section, we focus on the output buffer of a bottleneck link over which a single ABR connection runs (the cases for multiple ABR connections will be treated in Section 7). The system dynamics could be in either *equilibrium* or *transient* state. We derive the analytical expressions — which determine the performance measures such as maximum queue length, average throughput, and rate/queue length function oscillation periods — for both equilibrium and transient states. We also derive the expressions which can be used to compute the evolutions of rate/queue length functions.

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. In this state, $R(t)$ fluctuates around the link bandwidth. The fluctuation amplitude and period are determined by the rate-control parameters α, β, buffer capacity ξ, congestion detection thresholds Q_h, Q_l, and delays T_b, T_f. The use of credit control yields three different patterns for flow-controlled rate and queue-length functions, depending on the ranges of the flow-control parameters.

Pattern I: $\xi \geq Q_{max}$. Under this condition, the allocated buffer is large enough for the connection never to run out of credit, so no buffer congestion occurs, and hence, only the rate-control mechanism governs the system dynamics. Figure 5 (assuming ICR=0) illustrates the rate fluctuation and related queue length at the bottleneck link for this pattern. At time t_0, the rate reaches the link bandwidth μ, and the queue starts to build up after a delay of T_b. At time $t_0 + T_b + T_q$, the queue length reaches Q_h and bandwidth congestion is detected. After a forward delay of T_f, the source receives CI=1 feedback and its rate begins to decrease exponentially. The queue length reaches the peak as the rate drops back to the link bandwidth μ. When the rate falls below the link bandwidth, the queue length starts to decrease. After a time period of T_l, the queue length reaches Q_l, then the non-congestion condition is detected and sent back to the source. After a forward delay of T_f, the source rate starts to increase linearly. When the rate reaches the link bandwidth again, the system starts a new cycle of fluctuation. $R(t)$ fluctuates, with a linear-increase and an exponential-decrease alternating, around the bottleneck bandwidth.

Let R_{max} and R_{min} be the maximum rate and minimum rate, respectively, and Q_{max} be the maximum queue length, then we have

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

where $T_q = \sqrt{\frac{2Q_h}{\alpha}}$ is the time for the queue length to grow to Q_h from zero. For the convenience of presentation, we define

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

which is the time for the source rate to increase from μ to its maximum R_{max} by exercising linear rate-increase control. Then, the maximum queue length is expressed as

$$Q_{max} = \int_0^{T_{max}} \alpha t \; dt + \int_0^{T_d} (R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu) dt \tag{5.3}$$

where T_d is the time for the rate to drop from R_{max} back to μ, and is obtained, by letting $R(T_d) = \mu$, as:

$$T_d = -\frac{\Delta}{(1-\beta)} \; log \; \frac{\mu}{R_{max}}. \tag{5.4}$$

Then, we have

$$Q_{max} = \frac{\alpha}{2}(T_{max})^2 + \alpha\frac{\Delta}{(1-\beta)}(T_{max}) + \mu\frac{\Delta}{(1-\beta)} \; log \; \frac{\mu}{R_{max}} \tag{5.5}$$

Note that the queue becomes empty during the fluctuations in equilibrium state as the utilization is $<$ 100 %. T_l is the duration for the queue length to decrease from Q_{max} to Q_l, and thus, is defined by the following equation:

$$Q_{max} - Q_l = \int_0^{T_l} \mu(1 - e^{-(1-\beta)\frac{t}{\Delta}}) dt \tag{5.6}$$

So, T_l is the non-negative real root of the nonlinear equation:

$$e^{-(1-\beta)\frac{T_l}{\Delta}} + \left(\frac{1-\beta}{\Delta}\right) T_l - \left[\left(\frac{Q_{max} - Q_l}{\mu}\right) \left(\frac{1-\beta}{\Delta}\right) + 1 \right] = 0. \tag{5.7}$$

The minimum rate is then given as

$$R_{min} = \mu e^{-(1-\beta)\frac{(T_l + T_b + T_f)}{\Delta}}. \tag{5.8}$$

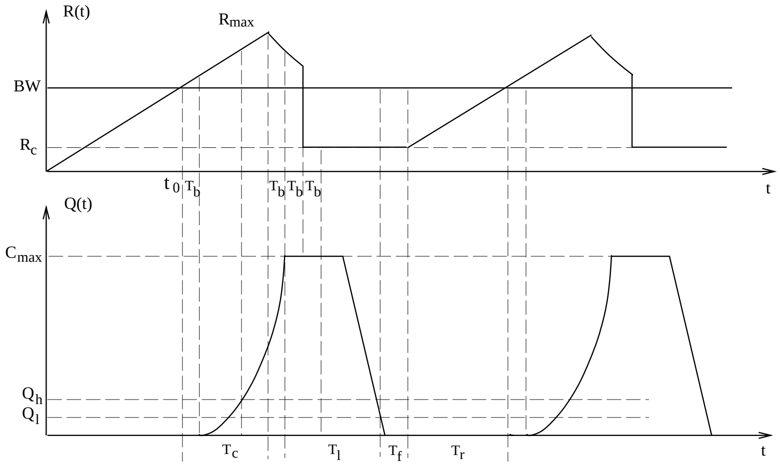


Figure 6: Dynamic behavior of $R(t)$ and $Q(t)$ for control pattern II.

By definition, the rate-fluctuation cycle is $T = T_q + T_d + T_l + 2\tau + T_r$, where $T_r = (\mu - R_{min})/\alpha$ is the time for $R(t)$ to grow from R_{min} to μ. Note that T contains two round-trip delays, which correspond to the two transitions of $R(t)$ (from linear to exponential and then back to linear).

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

$$\overline{R} \triangleq \frac{1}{T}\int_{t_0}^{t_0+T} R(t)dt = \frac{1}{T}\Big[\int_0^{T_{max}}(\mu+\alpha t)dt + \int_0^{T_e}(R_{max}e^{-(1-\beta)\frac{t}{\Delta}})dt + \int_0^{T_r}(R_{min}+\alpha t)dt\Big] \quad (5.9)$$

where $T_e = T_d + T_l + \tau$ is the time for exponential-decrease rate control within a cycle. Reducing the above equation, we obtain

$$\overline{R} = \frac{1}{T}\Big[\mu T_{max} + \frac{\alpha}{2}T_{max}^2 + R_{max}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) + T_r R_{min} + \frac{\alpha}{2}T_r^2\Big] \quad (5.10)$$

Pattern II: $\frac{\alpha}{2}(T_{max} - 2T_b)^2 < \xi < Q_{max}$. This is the case where the queue length would potentially grow beyond the allocated buffer size in the absence of credit control, but the allocated buffer size is still large enough for the rate to reach the maximum under the rate-increase control (i.e., the source still needs to partially exercise exponential-decrease rate control). The dynamic behaviors of $R(t)$ and $Q(t)$ for Pattern II are plotted in Figure 6. Since $R(t)$ can reach R_{max}, the rate-increase control dynamic remains the same as in Pattern I. However, when $Q(t)$ reaches the allocated buffer capacity ξ (buffer congestion), the bottleneck node immediately generates and sends a backward RM (with BN=1) cell directly to the source. After a backward delay of T_b, the source receives this buffer congestion information 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 queueing 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 (see Eqs. (5.15), (5.22)). To make a tradeoff between queueing delay and average throughput, we set:

$$R_c = \max\left\{\frac{\mu}{R_{max}}(2\mu - R_{max}),\quad MCR\right\}. \quad (5.11)$$

If $R_{max} \geq 2\mu$ then $R_c = MCR$; and if $R_{max} = \mu$ then $R_c = \mu$. With a delay of T_b after $R(t) = R_c$, the buffered cells start draining out at the rate of $(\mu - R_c)$. Over a time period of T_l, $Q(t)$ drops from ξ to

Q_l. When the non-congestion indication ($Q(t) = Q_l$) is detected at the bottleneck node and received by the source after a forward delay of T_f, $R(t)$ restarts a linear-increase from R_c. After a time interval of T_r, $R(t)$ reaches μ and a new cycle of rate fluctuation begins.

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 new parameter T_c, the time for $Q(t)$ to increase from 0 to the maximum buffer size allocated, ξ. The system dynamics belong to one of these three cases, depending on the range T_c falls in. In all of these three cases, R_{max} is the same as that derived for Pattern I and $Q_{max} = \xi$. So, we focus on the rate-decrease control.

Case 1: $\underline{T_{max} - 2T_b < T_c < T_{max}}$. Since $R(t)$ always takes T_b units of time to affect $Q(t)$ at the bottleneck node and buffer congestion ($C_{bal} = 0$) feedback also experiences a delay of T_b to change $R(t)$ at the source, we consider the T_c's range from $T_{max} - 2T_b$ to T_{max}. Since $T_c \leq T_{max}$ (i.e., $Q(t) = \xi$ is due only to the linear-increase rate control), we get $T_c = \sqrt{\frac{2\xi}{\alpha}}$. As ξ varies within the range from $\frac{\alpha}{2}(T_{max} - 2T_b)^2$ to $\frac{\alpha}{2}(T_{max})^2$, the duration of exponential-decrease rate-control grows from 0 to $2T_b$. The next key parameter for determining performance measures is T_l, the duration from the time when $Q(t)$ reaches ξ to the time when $Q(t)$ drops to Q_l. Based on the proposed control algorithms, we obtain

$$T_l \quad = \quad 2T_b + \frac{\theta + (\xi - Q_l)}{\mu - R_c} \tag{5.12}$$

where

$$\theta = \frac{\alpha}{2}(T_{max})^2 + R_{max}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) - \mu T_e - \xi \tag{5.13}$$

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 (see the credit scheme in Section 2). In Eq. (5.13), $T_e = (T_c + 2T_b) - T_{max}$ is the duration of exponential-decrease rate control for Case 1 and T_e varies from 0 to $2T_b$. Then, the fluctuation cycle of Case 1 is:

$$T \quad = \quad T_c + T_b + T_l + T_f + T_r \tag{5.14}$$

where $T_r = (\mu - R_c)/\alpha$ is the time for $R(t)$ to increase from R_c to μ.

Using its definition in Eq. (5.9), the average throughput for this case is derived as:

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

Case 2: $\underline{T_{max} < T_c < T_{max} + T_d - 2T_b \text{ (assume } T_d > \tau \text{ and } T_d \text{ as defined in Eq. (5.4))}}$. This case is an immediate sequel to Case 1 and T_c will now range from T_{max} to $T_{max} + T_d - 2T_b$. Since $T_c > T_{max}$, i.e., the $Q(t) = \xi$ status is contributed by both the linear-increase and exponential-decrease rate control, T_c consists of two parts, one from the linear rate part and the other from the exponential part: $T_c = T_{max} + T_\xi$, where T_ξ is the duration for the exponential part of $R(t)$ contributing toward $Q(t) = \xi$ and is defined by

$$\xi = \frac{\alpha}{2}(T_{max})^2 + \int_0^{T_\xi}(R_{max}e^{-(1-\beta)\frac{t}{\Delta}} - \mu)dt. \tag{5.16}$$

Thus, T_ξ is the non-negative real solution of a nonlinear equation:

$$e^{-(1-\beta)\frac{T_\xi}{\Delta}} + \left(\frac{\mu}{R_{max}}\right)\left(\frac{1-\beta}{\Delta}\right)T_\xi + \left[\left(\frac{\xi - \frac{\alpha}{2}(T_{max})^2}{R_{max}}\right)\left(\frac{1-\beta}{\Delta}\right) - 1\right] = 0. \tag{5.17}$$

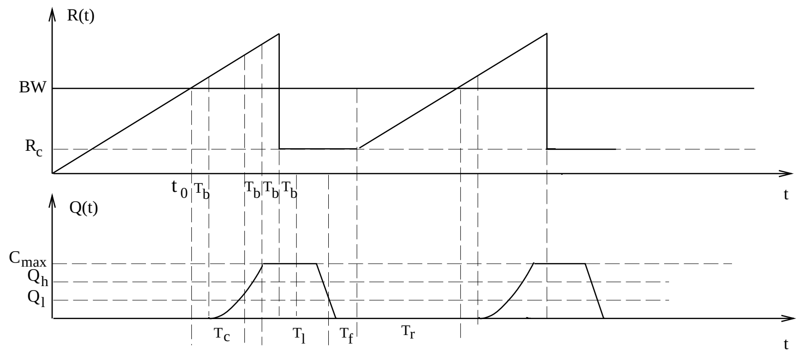


Figure 7: Dynamic behavior of $R(t)$ and $Q(t)$ for control pattern III.

Notice that if $\xi = \frac{\alpha}{2}(T_{max})^2$, the non-negative real solution to Eq. (5.17) is $T_\xi = 0$; this is expected since $\xi = \frac{\alpha}{2}(T_{max})^2$ implies that the exponential part does not contribute to $Q(t) = \xi$, and thus, by definition, T_ξ is supposed to be zero. Obviously, T_ξ ranges between 0 and $(T_d - 2T_b)$ for Case 2.

If β is close to 1, using the first-order series expansion (the higher-order series expansion should be applied if β is not sufficiently close to 1 or higher accuracy is required), T_ξ can be approximated by a simple expression

$$T_\xi \approx \frac{\Delta}{R_{max}(1-\beta)}\left[(R_{max}-\mu) - \sqrt{(R_{max}-\mu)^2 - 2R_{max}\left(\xi - \frac{\alpha}{2}T_{max}^2\right)\left(\frac{1-\beta}{\Delta}\right)}\right]. \tag{5.18}$$

Note that if $\xi = \frac{\alpha}{2}(T_{max})^2$, Eq. (5.18) gives $T_\xi = 0$, which is also consistent with the definition of T_ξ.

The expressions of T_l, θ, T, and $\overline{R}$ for Case 2 are then the same as Case 1 (given by Eqs. (5.12), (5.13), (5.14), and (5.15), respectively) except that in Case 2, $T_c = T_{max} + T_\xi$ and $T_e = T_\xi + 2T_b$ (T_e ranges from $2T_b$ to T_d).

Case 3: $\underline{T_{max} + T_d - 2T_b < T_c < T_{max} + T_d \text{ } (T_d \text{ is given by Eq. (5.4))}}$. This case completes Pattern II. As with Case 2, the status $Q(t) = \xi$ includes contributions from both the linear-increase and exponential-decrease of the rate control, thus leading to $T_c = T_{max} + T_\xi$.

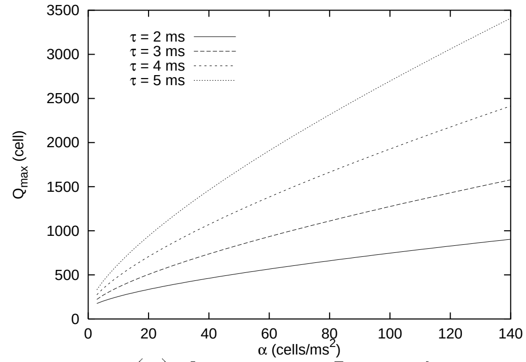
Most of the analytical results derived in Case 2 still hold for Case 3, including Eqs. (5.16),(5.17),(5.18) for the calculation of T_ξ. However, calculation of T_l is different, because T_c now ranges from T_{max} to $T_{max} + T_d$; T_ξ, by the definition of T_c for Case 3, will then vary from $T_d - 2T_b$ to T_d. To calculate T_l, we need to determine the "net" contribution of the "overshoot" cells generated in a delay interval of $2T_b$, during which the $R(t)$ curve intersects μ. Let θ (γ) denote the positive (negative) contribution — the area defined by the $R(t)$ curve above (below) μ. Then, the net contribution of overshoot cells is $\theta + \gamma$, which can be either positive or negative, depending on the time instant $R(t)$ intersects μ.

Similarly to derivation of Eq. (5.13), we get

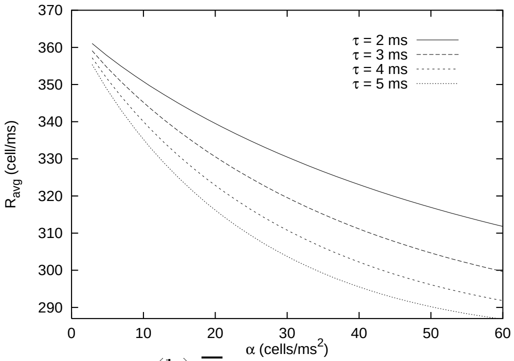
$$\theta = \frac{\alpha}{2}(T_{max})^2 + R_{max}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) - \mu T_d - \xi \tag{5.19}$$

$$\gamma = \mu\left[\frac{\Delta}{1-\beta}\left(1 - e^{-(1-\beta)\frac{2T_b-(T_d-T_\xi)}{\Delta}}\right) - [2T_b - (T_d - T_\xi)]\right]. \tag{5.20}$$

Note that if $T_\xi = T_d - 2T_b$, Eq. (5.20) gives $\gamma = 0$ and if $T_\xi = T_d$, Eq. (5.19) gives $\theta = 0$, both of which are consistent with the definitions of θ and γ.



(a) Q_{max} vs. α: Pattern I



(b) $\overline{R}$ vs. α: Pattern I

Figure 8: Q_{max} (maximum queue length) and $\overline{R}$ (average throughput) vs. α (rate of rate-increase).

The oscillation period can also be computed by Eq. (5.14), except that T_l in Eq. (5.14) is redefined for Case 3 as

$$T_l = 2T_b + \frac{(\theta + \gamma) + (\xi - Q_l)}{\mu - R_c}. \tag{5.21}$$

Using the definition (5.9), the average throughput for Case 3 is derived as:

$$\begin{aligned}\overline{R} &= \frac{1}{T}\Bigg[\mu T_{max} + \frac{\alpha}{2}T_{max}^2 + R_{max}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) \\ &\quad + \mu\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{2T_b-(T_d-T_\xi)}{\Delta}}\right) + (T_l - T_b + T_f)R_c + T_r R_c + \frac{\alpha}{2}T_r^2\Bigg].\end{aligned} \tag{5.22}$$

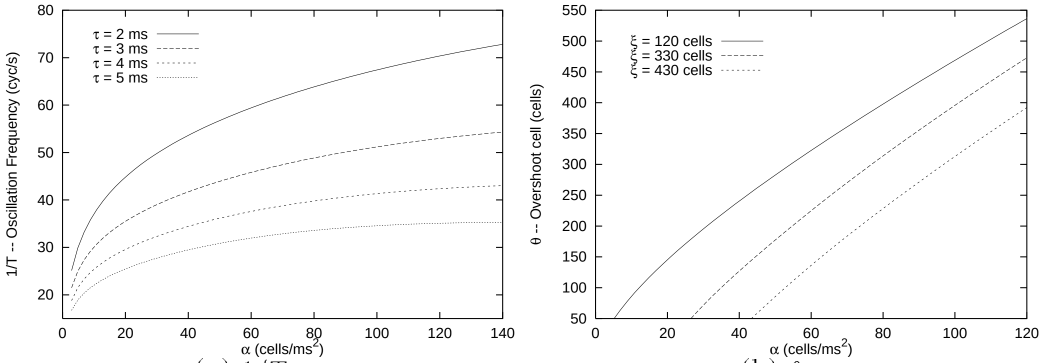
Pattern III: $0 < \xi < \frac{\alpha}{2}(T_{max} - 2T_b)^2$. This pattern is similar to Pattern II in that buffer congestion control is also exercised. However, in this pattern, for given α, the allocated buffer size is so small that the source rate cannot reach R_{max} specified by the rate control algorithm and the exponential-decrease rate control is never exercised. Figure 7 illustrates the dynamic behavior for this pattern. When the source receives $Q(t) = \xi$ ($C_{bal} = 0$) feedback, its current value of $R(t)$ is reduced to an appropriate smaller value, R_c. T_b time units after the point $R(t) = R_c$, the buffered cells begin to "dissipate" at rate $\mu - R_c$. After a time interval of T_l, $Q(t)$ falls down to Q_l and the non-congestion indication ($Q(t) = Q_l$) is detected. When the source receives the non-congestion feedback, T_f after the point $Q(t) = Q_l$, it restarts increasing $R(t)$ from R_c. After the linear-increase of rate for T_r time units, $R(t)$ will reach μ and the system will then start the next cycle of fluctuation.

Using the same T_c as defined in Pattern II's Case 1, we get $T_c = \sqrt{\frac{2\xi}{\alpha}}$. Since R_{max} defined by Eq. (5.1) cannot be reached, the new maximum rate, R'_{max}, for Pattern III is recalculated as $R'_{max} = \mu + \alpha(T_c + 2T_b)$, where $R'_{max} \le R_{max}$.

Also, using the same definitions of θ, T_l, and Eq. (5.12) as in Pattern II, we obtain

$$\theta = \frac{\alpha}{2}(T_c + 2T_b)^2 - \xi \tag{5.23}$$

$$T_l = 2T_b + \frac{\frac{\alpha}{2}(T_c + 2T_b)^2 - Q_l}{\mu - R_c}. \tag{5.24}$$



(a) $1/T$ vs. α (b) θ vs. α

Figure 9: $1/T$ (oscillation frequency) and θ (overshoot cells) vs. α (rate of rate-increase).

Finally, by substituting these T_c and T_l into Eqs. (5.14) and (5.9), we obtain the average throughput for Pattern III:

$$\overline{R} \;=\; \frac{1}{T}\left[\;\mu(T_c+2T_b)+\frac{\alpha}{2}(T_c+2T_b)^2+(T_l-T_b+T_f)R_c+T_rR_c+\frac{\alpha}{2}T_r^2\right]. \tag{5.25}$$

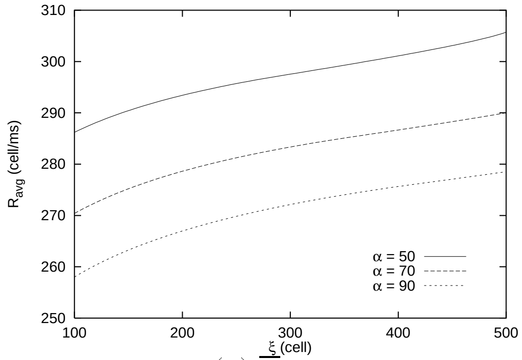
5.2 Numerical Evaluation of Equilibrium-State Performance

Using the analytical results derived thus far, we now present numerical analyses of the equilibrium-state performance. The bottleneck link bandwidth is set at $\mu = 155$ Mbps and we assume $T_b = T_f = 1$ ms (i.e., the bottleneck is identified at the farthest node, the worst case in view of feedback delay), 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 $R_0 = \mu$ (since we are concerned with the equilibrium state).

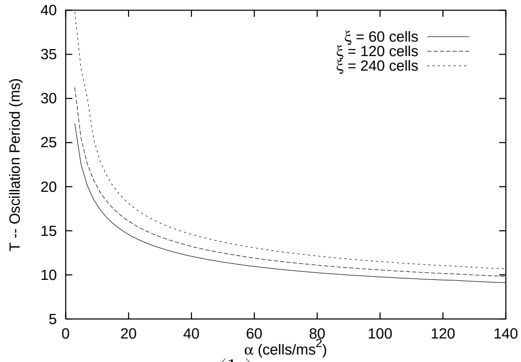
To balance the increase and decrease speeds of $R(t)$ and ensure that the average of the offered traffic load does not grow beyond the bottleneck bandwidth, we set $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$ throughout the rest of the paper. Since α represents $R(t)$'s increase rate and $\left(\frac{1-\beta}{\Delta}\right)$ determines $R(t)$'s decrease speed, setting $\alpha/\left(\frac{1-\beta}{\Delta}\right) = \mu$ means that near the point ($R(t) = \mu$, $Q(t) = 0$), the rate of $R(t)$'s increase is equal to the rate at which it decreases [18]. In addition, setting $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$ confines us to a simpler scenario where we only need to control one parameter α, instead of both α and β.

Performance Analysis for Pattern I ($\xi \geq Q_{max}$): First, we consider how the rate-increase parameter α affects the maximum queue length Q_{max}. Figure 8(a) plots Q_{max} while varying α for different values of round-trip delay τ. As expected, 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. Figure 8(b) plots the average throughput $\overline{R}$ against α with different τ values. $\overline{R}$ is found to decrease monotonically as α and τ increase, and $\overline{R}$ to decrease faster for a larger τ. In Figure 9(a), the oscillation frequency $1/T$ is plotted while increasing α for different τ's. The oscillation frequency gets higher as α increases and τ decreases. 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 average throughput.

Performance Analysis for Pattern II and III ($0 < \xi < Q_{max}$): First, we consider how the rate-control parameter α influences θ, the number of cells queued due to buffer congestion. In Figure 9(b), θ is plotted against α with different values of buffer capacity ξ ($< Q_{max}$). As expected, θ increases



(a) $\overline{R}$ vs. ξ



(b) T vs. α

Figure 10: Average Throughput vs. Buffer Capacity and Oscillation Period vs. Rate of Rate-Increase.

monotonically as α increases, but for given α, θ decreases as ξ increases. Figure 10(a) plots the average throughput $\overline{R}$ while increasing buffer capacity ξ for different values of α. $\overline{R}$ is found to monotonically increase as ξ increases, but for given ξ, $\overline{R}$ to decrease as α increases. The relationship between the oscillation period T and the rate-increase parameter α is illustrated in Figure 10(b). As in Pattern I, T decreases as α increases and ξ decreases. T is also found to decrease abruptly for smaller α's and slow down the decrease when α gets larger.

Performance Comparison among Three Control Patterns: To make a direct performance comparison among Patterns I, II, and III, in Figure 11 $\overline{R}/\mu$'s are plotted against α with different values of ξ, which correspond to different control patterns. We observe:

1. For any given α, the equilibrium state governed by Pattern I represents the optimal equilibrium state in terms of average throughput, queueing delay, and delay variation. Thus, we define control Pattern I as the optimal control pattern/mode, and the equilibrium state under control of Pattern I as the optimal equilibrium state.
2. For given ξ, $\overline{R}$ monotonically decreases as α increases for all three patterns.
3. $\overline{R}$ of Pattern II and III with a smaller ξ decays faster as α increases.
4. 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.

We can summarize our observations on equilibrium-state performance analysis as follows.

- α and ξ have major impacts on equilibrium-state performance
  - A larger α results in a lower average throughput $\overline{R}$.
  - For given α, the average throughput $\overline{R}$ increases with ξ.
  - Bandwidth utilization is bounded by the curve $\xi \geq Q_{max}$, which provides information on optimal buffer allocation for given α.
  - The larger α, the more sensitive to ξ the average throughput $\overline{R}$ is.
- A smaller α leads to better equilibrium-state performance.
- Pattern I is optimal in terms of average throughput, bandwidth utilization, and delay.
- For given ξ, α is critical for the system to operate under the optimal pattern.
- Q_{max} is proportional to α.

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. But α should not be too small since a small α degrades transient-state performance.

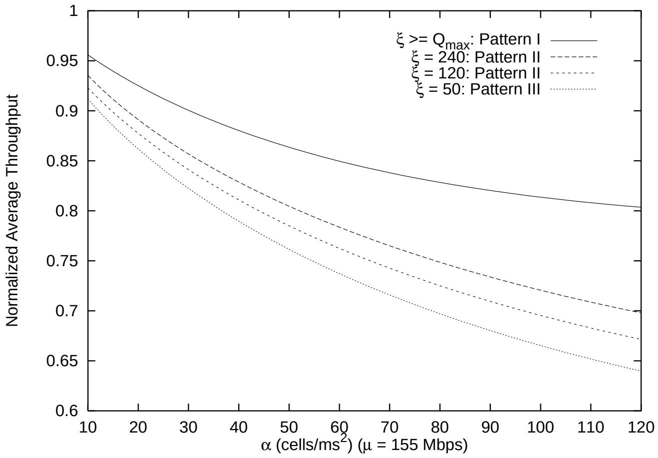


Figure 11: Control Pattern Comparison: $\overline{R}/\mu$ vs. α with different ξ.

6 Second-Order Rate Control and Transient Performance Analysis

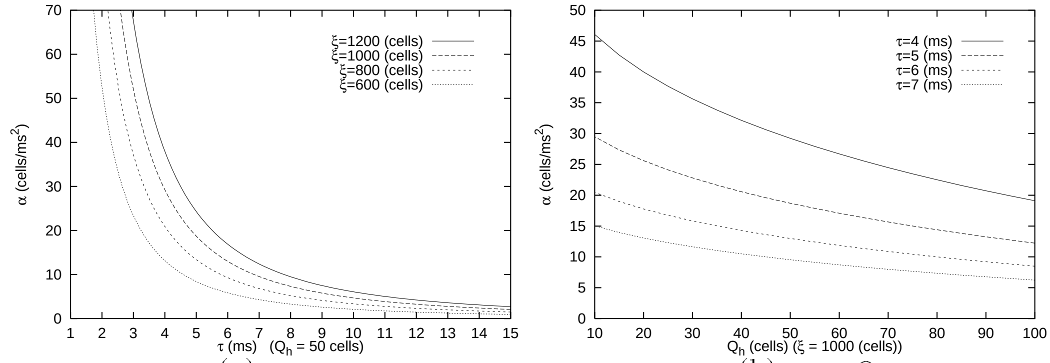
Now, let's turn our attention to the system dynamics and performance in transient state. We begin with the transient state rate-control algorithm.

6.1 The Second-Order Rate Control Algorithm and its Properties and Performance

As shown above, flow control Pattern I is optimal in equilibrium state in terms of average throughput, bandwidth utilization, and delay. We therefore define flow control Pattern I ($\xi \geq Q_{max}$) as the *optimal equilibrium state*, where the rate function converges to a small neighborhood of the target bandwidth, and the queue-length function fluctuates under the buffer capacity (zero overshoot cell). The goal of our transient-state rate-control is to drive the system from any initial state into the optimal equilibrium state as quickly as possible. Thus, the *transient state* is defined as a state between any initial state and an optimal equilibrium state.

As discussed in Section 5, exercising increase or decrease control over $R(t)$ alone is not effective enough to have the system converge to the optimal equilibrium state for given buffer capacity ξ. This is because 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}. According to the analytical results obtained in Section 5.1, $Q_{max} = Q_{max}(\alpha, \beta)$ is a function of α and β, and thus we can adjust the rate parameter α (β is also adjusted by setting $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$ accordingly) to control Q_{max}. Since the control is over $\alpha = \frac{dR(t)}{dt}$, we call it the second-order control over $R(t)$ which provides one more dimension to control the dynamics of the proposed flow-control system. In addition, there are other reasons why α needs to be adjusted dynamically. In a real network, backward delay T_b, forward delay T_f, and hence, the round-trip delay τ vary with time; for instance, T_b and T_f are functions of the bottleneck location. Thus, according to the analytical results in the last section, keeping Q_{max} at a given level requires α to vary with time. Moreover, τ is usually not known *a priori* at the source, and thus, the source needs to adjust α to match τ upon receiving a feedback that experiences the "true" round-trip delay τ. Likewise, keeping $\overline{R}$ at a certain level requires α to vary with time.

Theorem 1 that follows formally states why the second-order rate control can drive the system from any initial transient state to an optimal equilibrium state.



(a) α vs. τ (b) α vs. Q_h

Figure 12: Rate-increase parameter vs. RM-cell round trip delay and buffer-occupancy threshold.

Theorem 1. *Consider an ABR connection flow-controlled by the proposed scheme with buffer occupancy threshold equal to Q_h, rate-increase parameter equal to α, buffer capacity equal to ξ, and RM-cell round-trip delay equal to $\tau > 0$.* ***If*** $\sqrt{\alpha} \leq \frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}$, ***then*** *(1) the ABR connection is guaranteed to be in an optimal equilibrium state, and (2) the maximum buffer occupancy is upper bounded as $Q_{max} \leq (\tau\sqrt{\alpha} + \sqrt{2Q_h})^2$.* □

Basically, the theorem reveals the following facts. For given flow-control system parameters ξ, τ, and Q_h, a system optimal equilibrium state can always be achieved by controlling rate parameter α. This property of the second-order rate control is explicitly described by a sufficient condition which guarantees the existence of the optimal value of α and gives a simple analytical expression to estimate the optimal α for given ξ, τ, and Q_h. In addition, the upper bound of Q_{max} is found to be a function of flow-control parameters under the optimal equilibrium state, and the upper bound of Q_{max} is more sensitive to τ than to α. To quantitatively study the relationship (under the optimal equilibrium state) among the flow-control parameters described by Theorem 1, in Figure 12(a) we plot α against τ by letting $\sqrt{\alpha} = \frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}$ with ξ varying and $Q_h = 50$ cells. As expected, α decreases as τ increases since slow feedback requires a gentle rate increase to keep buffer occupancy under a given level. We also observe that for given τ, a large ξ can sustain an aggressive α, because a larger ξ allows for a larger buffer occupancy.

Theorem 1 also indicates that the buffer occupancy threshold Q_h has a direct impact on Q_{max}, which is analytically described in the theorem condition and the upper-bound of Q_{max}. This fact enables the network designers to compute Q_h more formally, as compared to using heuristics as in previous rate-based schemes. Based on the theorem condition, α is plotted against Q_h in Figure 12(b) with different τ in the optimal equilibrium state. α is found to be a monotonically decreasing function of Q_h, which is expected since a large Q_h delays the congestion signal to be sent back to the source, and thus requires a less-aggressive rate increase to ensure $Q_{max} \leq \xi$. We also observe that for given α, a larger τ corresponds to a smaller Q_h in the optimal state. Therefore, Q_{max} can also be controlled by dynamically adjusting Q_h. However, dynamical control over Q_h poses implementation difficulties since Q_h's are distributed over all switches. Nevertheless, Theorem 1 provides an analytical expression, by which Q_h can be calculated accurately for the given ξ, τ, and the desired range of α.

If the RM-cell round-trip delay τ is not a constant or unknown *a priori*, then we can rearrange the sufficient condition to be $\tau\sqrt{\alpha} \leq \sqrt{\xi} - \sqrt{2Q_h}$, and Theorem 1 still holds. Thus, this new version of the theorem shows that Q_{max} is subjected to $\tau\sqrt{\alpha}$, and can be controlled by decreasing/increasing α in response to the variation of τ. However, here we only concentrate on how to reduce α to ensure the maximum queue length does not grow beyond a given buffer capacity while achieving a good transient response. A more general algorithm for the second-order rate control is detailed in [21], where we proposed an optimal α-control law for both decreasing and increasing the rate parameter α.

The control rule for reducing α is described by a discrete function, as it is exercised only at time instants when the buffer congestion occurs. Let α_0 be the initial source rate-increase parameter. Application of this control rule n times will yield a sequence $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$. For a good transient response, we propose to use an exponential control rule which is defined by

$$\alpha_n = e^{-\lambda n}\alpha_0 \quad (0 < \lambda). \tag{6.1}$$

where λ specifies the speed of reducing α.

Note that α should not be reduced further as long as $Q_{max}(\alpha_n) \leq \xi$, where Q_{max} is a function of α as illustrated in Eq. (5.5), since too small a value of α will slow down the transient system response, or even disable the capability of grabbing the spare/unused bandwidth created by other idle VCs. So, the source should stop execution of the reducing rule as soon as α_n reaches its optimal value α^* which is defined below.

Definition 1. *The optimal value of the rate parameter α which corresponds to an optimal equilibrium state under the second-order rate control for a given initial rate parameter α_0 is defined by*

$$\alpha^* \triangleq \max_{i \in \{1,2,3,\ldots\}} \{\alpha_i \mid Q^{(i)}_{max} \leq \xi\}, \tag{6.2}$$

where $Q^{(i)}_{max} = Q_{max}(\alpha_i)$ for $\alpha_i = e^{-\lambda}\alpha_{i-1}$, and $\alpha_0 > \alpha_\xi$ which is the rate parameter corresponding to $Q_{max}(\alpha_\xi) = \xi$, the buffer capacity of the bottleneck node.

The performance of the second-order rate control is characterized by the speed of convergence to an optimal equilibrium state and closeness of $Q_{max}(\alpha^*)$ to the bottleneck-node target buffer occupancy at the equilibrium-state. The former is measured by the number of the transient-state cycles, denoted by N, that a flow-control system experiences from any initial state to an optimal equilibrium state. The latter is evaluated by the difference between α^* and α_ξ. For fast convergence, a smaller N is desirable, while a larger α^* (closer to α_ξ) is favorable in terms of buffer utilization and responsiveness in grabbing the unused bandwidth. The values of N and α^* both depend upon the second-order rate control parameter λ and other flow-control parameters. Theorem 2 given below characterizes both N and α^* by providing their worst-case bounds as the functions of flow-control parameters.

Theorem 2. *Consider an ABR connection flow-controlled by the proposed scheme with buffer occupancy threshold equal to Q_h, rate increase parameter equal to α, buffer capacity equal to ξ, and RM-cell round-trip delay equal to $\tau > 0$.* ***If*** *the second-order rate control law given in Eq. (6.1) is applied to this ABR connection with the initial rate-control parameter $\alpha_0 > \alpha_\xi$ for a given $\lambda > 0$,* ***then*** *(1) the number of transient-state cycles is upper bounded as*

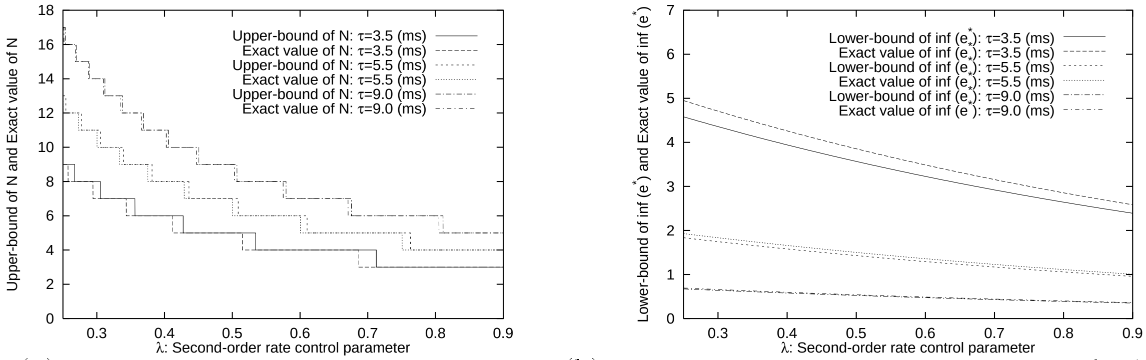
$$N \leq \left\lceil \frac{2}{\lambda} \, log \, \left(\frac{\tau\sqrt{\alpha_0}}{\sqrt{\xi} - \sqrt{2Q_h}} \right) \right\rceil; \tag{6.3}$$

(2) the rate parameter for the resultant optimal equilibrium state is lower bounded as

$$\inf_{\forall \alpha_0 > \alpha_\xi} \{\alpha^*(\alpha_0)\} \geq e^{-\lambda} \left(\frac{\sqrt{\xi} - \sqrt{2Q_h}}{\tau} \right)^2. \tag{6.4}$$

□

The proof is detailed in the appendix. Claim (1) of Theorem 2 reveals that the upper bound of N is inversely proportional to λ, the second-order rate control parameter, for the given values of other



(a) Upper-bound and exact value of N vs. λ (b) Lower-bound and exact value of $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ vs. λ

Figure 13: Worst-case bounds and exact values of N and $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ vs. λ

parameters. This makes sense since λ represents the speed of reducing α. Moreover, for a given λ, N tends to increase with τ, which is also expected, since a large τ requires a small α^* in an optimal state. In general, a large λ accelerates convergence to the optimal state. On the other hand, the claim (2) of Theorem 2 identifies a lower bound for $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ which indirectly specifies the maximum (worst-case) distance between α^* and α_ξ since

$$\sup_{\forall\alpha_0>\alpha_\xi}\{\alpha_\xi-\alpha^*(\alpha_0)\}=\alpha_\xi-\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}\le\alpha_\xi-e^{-\lambda}\left(\frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}\right)^2. \qquad (6.5)$$

So, a larger lower-bound for $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ is desired. Note that, unlike in claim (1), the worst-case bound of Eq. (6.4) is somewhat "conservative" in the sense that α^* may potentially take any value over the interval $\left(e^{-\lambda}\left(\frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}\right)^2,\alpha_\xi\right]$ depending on the distribution of α_0 for the given λ. But a larger lower-bound for $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ will result in a larger α^* on the average. The claim (2) of Theorem 2 indicates that the lower-bound of $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ increases as λ decays for the given values of other parameters. Thus, a small λ is desired to increase buffer utilization and responsiveness in grabbing unused bandwidth on the average. However, this introduces a trade-off on selection of λ since a large λ favors a small N, whereas a small λ favors a large α^* on the average. The analytical relationships derived in Theorem 2 not only explicitly describe λ's impact on N and α^*, but also provide a clear insight on how all the other rate parameters affect the performance of the second-order rate control. In Figure 13, the worst-case bounds of N and $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ are plotted against λ for different τ values with $\xi=300$ cells, $\alpha_0=50$ cells/ms^2, and $Q_h=40$ cells, which all confirm our analysis discussed above.

In order to quantitatively evaluate the tightness of the derived worst-case bounds, the exact values of N and $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$ are also plotted against λ with different τ for $\mu=367$ cells/ms and $\xi=300$ cells in Figure 13. These bounds are found to be very tight when τ is large, because a large τ requires a small α for an optimal equilibrium state according to Theorem 1, which leads to $\mu<R_{max}\ll 2\mu$ (the tighter bound condition) for a given ξ. Thus, the derived bounds can very well approximate the performance of the second-order rate control since $\mu<R_{max}<<2\mu$ (due to small α) is the typical operating regime of the proposed scheme for a given ξ. Therefore, Theorem 2 provides the network designer with a means to accurately estimate the performance of the second-order rate control without seeking the exact closed-form expressions for $\alpha_\xi=Q^{-1}_{max}(\xi)$, N, and $\inf_{\forall\alpha_0>\alpha_\xi}\{\alpha^*(\alpha_0)\}$, which are impossible to obtain. Another interesting fact revealed by the above two theorems is that Q_{max} is virtually independent of the bottleneck bandwidth μ when $R_0<\mu$ since neither the optimal state condition nor the worst-case bounds contain μ. This should not surprise us since it is the relative difference between $R(t)$ and μ, instead of the absolute value of μ, that mainly determines Q_{max}.

6.2 Analytical Solutions and Analysis for Transient-State Performance

For the transient-state analysis, we assume $\alpha_0 \geq \alpha^*$, and focus on the first-cycle dynamic behavior with initial rate $R_0 > \mu$, because the analytical expressions for the subsequent cycles or the case of $R_0 \leq \mu$ during the first cycle are the same as those for equilibrium state. Like the equilibrium-state analysis, transient state flow-control also divides control into Patterns I, II, and III, 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 (the buffer congestion control applied), our analysis will focus on these two patterns. Let R_{peak} and Q_{peak} be the transient-state maximum rate and queue length, respectively. Notice that for Patterns II and III, when $Q_{peak} > \xi$, $R(t)$ restarts rate-increase from $R_c = \max\left\{\frac{\mu}{R_{peak}}(2\mu - R_{peak}),\ \ MCR\right\}$ with a smaller increase rate of $\alpha e^{-\lambda}$, instead of α.

We will give only Pattern II/III's analytical expressions for the performance measures of interest, and omit the derivation of these expressions and all the other performance measures (including Pattern I's) due to space limit.

Pattern II: Buffer congestion control is exercised and $R(t)$ reaches R_{peak}.

(1) Average throughput for Cases 1 and 2:

$$\overline{R} = \frac{1}{T}\left[R_0 T_{peak} + \frac{\alpha}{2}T_{peak}^2 + R_{peak}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right) + (T_l - T_b + T_f)R_c + T_r R_c + \frac{1}{2}\alpha e^{-\lambda} T_r^2\right] \quad (6.6)$$

where

$$T_{peak} = \tau + \frac{1}{\alpha}\left[-(R_0-\mu) + \sqrt{(R_0-\mu)^2 + 2\alpha Q_h}\right], \qquad R_{peak} = R_0 + \alpha T_{peak},$$

$$T_r = \frac{(\mu - R_c)}{\alpha e^{-\lambda}}, \quad (6.7)$$

$$T_l = 2T_b + \frac{\theta + (\xi - Q_l)}{\mu - R_c}, \quad (6.8)$$

$$\theta = (R_0 - \mu)T_{peak} + \frac{\alpha}{2}(T_{peak})^2 + R_{peak}\left(\frac{\Delta}{1-\beta}\right)\left[1 - e^{-(1-\beta)\frac{T_e}{\Delta}}\right] - \mu T_e - \xi,$$

$$T_e = \begin{cases} \frac{1}{\alpha}\left[-(R_0-\mu) + \sqrt{(R_0-\mu)^2 + 2\alpha\xi}\right] + 2T_b - T_{peak}; & \text{for Case 1} \\ T_\xi + 2T_b; & \text{for Case 2} \end{cases}$$

and T_ξ is the non-negative real root of the following non-linear equation

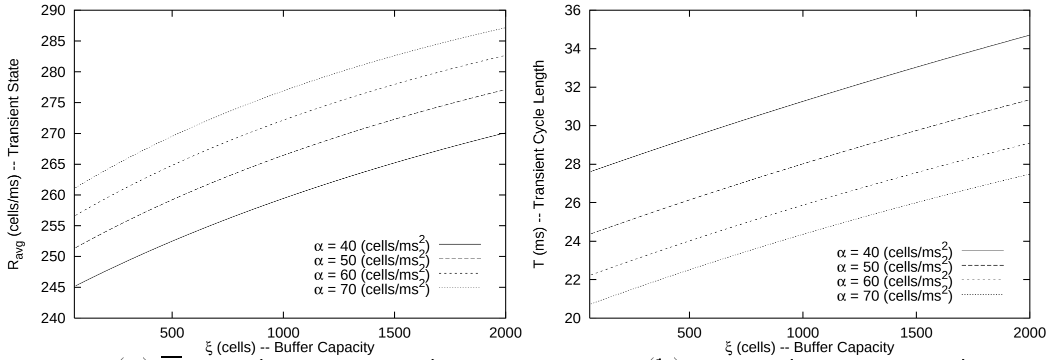
$$e^{-(1-\beta)\frac{T_\xi}{\Delta}} + \left(\frac{\mu}{R_{peak}}\right)\left(\frac{1-\beta}{\Delta}\right)T_\xi + \left[\left(\frac{\xi - \frac{\alpha}{2}(T_{peak})^2 - (R_0-\mu)T_{peak}}{R_{peak}}\right)\left(\frac{1-\beta}{\Delta}\right) - 1\right] = 0. \quad (6.9)$$

(2) Transient cycle length for Cases 1 and 2:

$$T = \tau + T_c + T_l + T_r \quad (6.10)$$

where T_r is given by Eq. (6.7), T_l is given by Eq. (6.8), and

$$T_c = \begin{cases} \frac{1}{\alpha}\left[-(R_0-\mu) + \sqrt{(R_0-\mu)^2 + 2\alpha\xi}\right]; & \text{for Case 1} \\ T_{peak} + T_\xi; & \text{for Case 2} \end{cases} \quad (6.11)$$



(a) $\overline{R}$ vs. ξ (Transient state) (b) T vs. ξ (Transient state)

Figure 14: $\overline{R}$ (average throughput) and T (transient period) vs. ξ (buffer capacity).

(3) Average throughput for Case 3:

$$\begin{aligned}\overline{R} \;&=\; \frac{1}{T}\Bigg[R_0 T_{peak} + \frac{\alpha}{2}T_{peak}^2 + R_{peak}\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) \\ &+ \mu\left(\frac{\Delta}{1-\beta}\right)\left(1 - e^{-(1-\beta)\frac{2T_b-(T_d-T_\xi)}{\Delta}}\right) + (T_l - T_b + T_f)R_c + T_r R_c + \frac{1}{2}\alpha e^{-\lambda}T_r^2\Bigg] \qquad (6.12)\end{aligned}$$

where all variables have the same definitions as in Cases 1 and 2, except that

$$T_d \;=\; -\frac{\Delta}{(1-\beta)}\; log\; \frac{\mu}{R_{peak}} \qquad (6.13)$$

$$\begin{aligned}T_l \;&=\; 2T_b + \frac{1}{\mu - R_c}\Bigg\{(R_0-\mu)T_{peak} + \frac{\alpha}{2}(T_{peak})^2 + \left(\frac{\Delta}{1-\beta}\right)\Bigg[R_{peak}\left(1 - e^{-(1-\beta)\frac{T_d}{\Delta}}\right) \\ &+ \mu\left(1 - e^{-(1-\beta)\frac{2T_b-(T_d-T_\xi)}{\Delta}}\right)\Bigg] - \mu(2T_b + T_\xi) - Q_l\Bigg\} \qquad (6.14)\end{aligned}$$

(4) Transient cycle length for Case 3:

$$T = \tau + T_c + T_l + T_r \qquad (6.15)$$

where T_c is defined by Case 2 of Eq. (6.11) and T_l is defined by Eq. (6.14).

Pattern III: Buffer congestion control is exercised and $R(t)$ does not reach R_{peak}.

(1) Average throughput:

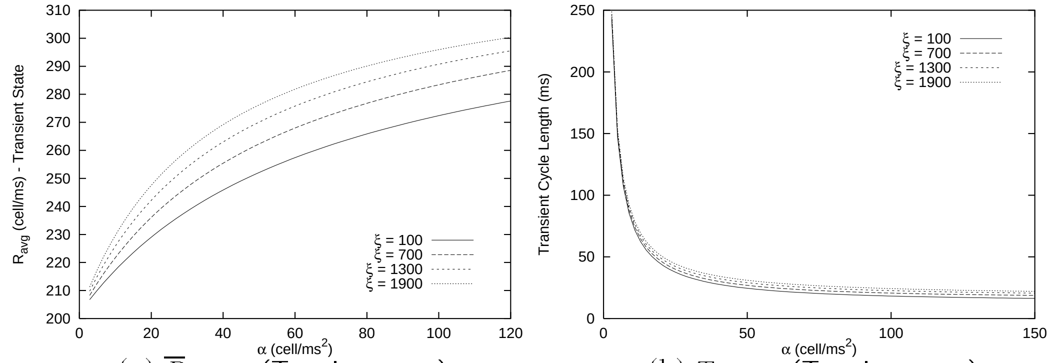
$$\overline{R} \;=\; \frac{1}{T}\Bigg[\; R_0(T_c+2T_b) + \frac{\alpha}{2}(T_c+2T_b)^2 + (T_l - T_b + T_f)R_c + T_r R_c + \frac{\alpha}{2}e^{-\lambda}T_r^2\Bigg] \qquad (6.16)$$

where T_c is given by Case 1 of Eq. (6.11), T_r is given by Eq. (6.7), and

$$T_l \;=\; 2T_b + \frac{1}{\mu - R_c}\Bigg[(R_0-\mu)(T_c+2T_b) + \frac{\alpha}{2}(T_c+2T_b)^2 - Q_l\Bigg]. \qquad (6.17)$$

(2) Transient cycle length: (all variables are defined as in Pattern III's average throughput)

$$T \;=\; \tau + T_c + T_l + T_r. \qquad (6.18)$$



(a) $\overline{R}$ vs. α (Transient state) (b) T vs. α (Transient state)

Figure 15: $\overline{R}$ (transient average throughput) and T (transient period) vs. α (rate of rate–increase) .

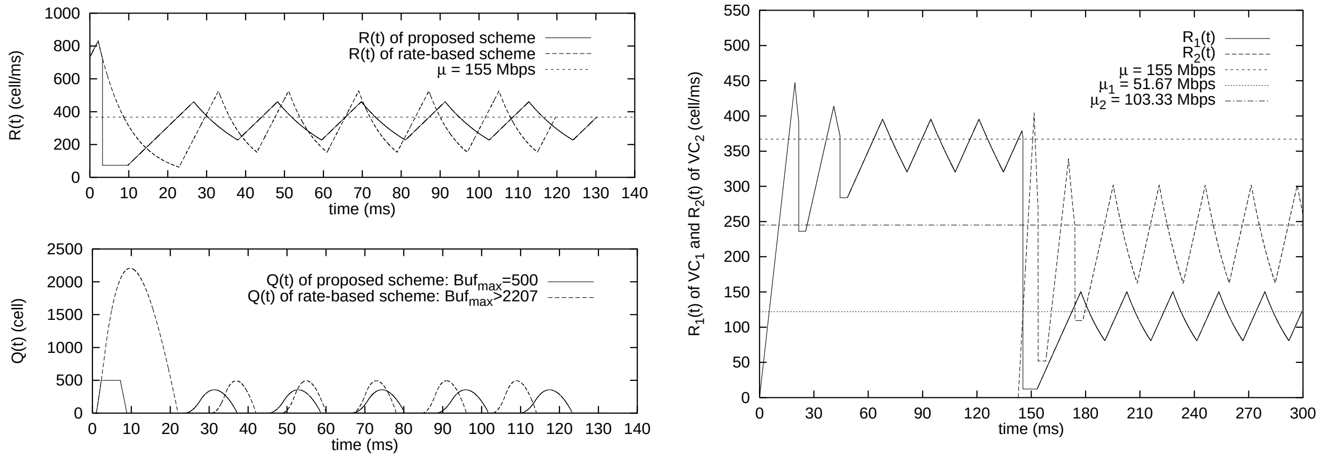
Using the analytical solutions derived above, we present numerical results on how system parameters affect the transient-state performance. The network condition remains the same as in Section 5.2. But we use $R_0 = 4\mu$ and $\lambda = log\ 2$[1] here. We first consider how buffer capacity ξ influences the average throughput $\overline{R}$ and the first transient-cycle length T. Figure 14 indicates that for given α a larger buffer capacity ξ leads to a higher throughput $\overline{R}$, but also suffers from a longer transient-cycle period T. This poses a trade-off between throughput and transient-cycle length. Figure 15 shows how α affects average throughput $\overline{R}$ and transient-cycle length T for different buffer capacity ξ. We observe that for 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 the source should start sending data with a larger initial rate-control parameter α_0, but make α smaller as system converges to the optimal equilibrium state. This is consistent with our control algorithms.

7 Dynamics of Multiple ABR Connections

To analyze the performance of the proposed scheme for the multiple-connection case, let's consider a scenario where M flow-controlled VCs share a common bottleneck and $J \leq M$ VCs are active. The parameters describing VC_i in this multi-connection system are given below.

μ_i:	$\triangleq \frac{MCR_i}{\sum_{j=1}^{J} MCR_j}\mu$ is bandwidth share for VC_i where μ is total bottleneck link bandwidth
ξ_i:	$\triangleq \frac{MCR_i}{\sum_{j=1}^{M} MCR_j}\xi$ is buffer share for VC_i where ξ is total buffer capacity at bottleneck node
$Q_h^{(i)}$:	$\triangleq \frac{MCR_i}{\sum_{j=1}^{M} MCR_j}Q_h$ is high buffer-threshold share for VC_i where Q_h is total high buffer-threshold
$Q_l^{(i)}$:	$\triangleq \frac{MCR_i}{\sum_{j=1}^{M} MCR_j}Q_l$ is low buffer-threshold share for VC_i where Q_l is total low buffer-threshold
MCR_i:	MCR for VC_i where we assume $MCR_i > 0, \forall i \in \{1, 2, \ldots, M\}$
$Q_i(t)$:	Queue length at the bottleneck node for VC_i
$R_i(t)$:	Source rate 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.

[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.



(a) Comparison of rate-based and proposed schemes. (b) Evolution of $R_1(t)$ and $R_2(t)$.

Figure 16: Comparison of proposed schemes with existing schemes.

The bottleneck node allocates its total buffer capacity ξ to M existing VCs, each with a buffer proportional to its MCR at the time of connection setup. Using the (work-conserving) WRR scheduler at the switch, $J \leq M$ *active* VCs are served in a round-robin fashion, but the service rates are proportional to their MCRs. So, the bottleneck bandwidth μ is dynamically shared by J active VCs. As a result, all active VCs interact with each other directly through the shared bandwidth μ. 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 Sections 5 and 6 can be applied to the multiple-connection case with the target bandwidth and buffer capacity substituted by their shares.

Now, we describe two examples to compare the proposed scheme with rate and credit schemes in terms of buffer requirement, bandwidth guarantee, fairness, and average throughput.

7.1 Buffer Requirement and Average Throughput

In the first example, we consider a case where there are $M = 4$ identical ABR VCs with the parameters: $R_0^{(i)} = 183.5$ cells/ms, $MCR_i = 18.35$ cells/ms, $T_b^{(i)} = T_f^{(i)} = 1$ ms (so $\tau_i = 2$ ms), $\Delta_i = 1$ ms, $\alpha_0^{(i)} = 11.45$ cells/ms^2 ($i = 1, 2, 3, 4$), $Q_h = 50$ cells, and $Q_l = 25$ cells. We assume that 4 VCs start sending cells at the same time over the bottleneck link with a bandwidth of $\mu = 367$ cells/ms (155 $Mbps$). These connection and network conditions are shared by both the rate and proposed schemes, which will be described below.

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, which describe the rate scheme, we obtained the evolution functions of $R(t) = 4R_i(t)$ and $Q(t)$, as shown in Figure 16(a). We observe that in transient state, there is a large queue build-up, $Q_{peak} = 2207.5$ cells. But in equilibrium state, Q_{max} is just 493 cells, about 1/5 of Q_{peak}. If lossless transmission is required, the buffer size at the switch must be larger than 2207 cells 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 total average throughput in equilibrium state is 319.35 cells/ms (or $\overline{R}/\mu = 0.87$).

With the proposed scheme, we assume that the bottleneck switch has a given buffer capacity of $\xi = 500$ cells. Then, each VC has $\xi_i = 125$ cells, $Q_h^{(i)} = 12.5$ cells, and $Q_l^{(i)} = 6.25$ cells since all VCs have the same MCR_i. The 4 VCs each have their own output queue at the bottleneck switch, but evenly share the link bandwidth μ. Using the equations derived for Pattern I, II, and III which characterize the proposed scheme, we computed the evolution functions 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 16(a), $R(t)$ just experiences one cycle of transient state with $\alpha_0^{(i)} = 11.45$ cells/ms^2 and then enters the equilibrium state with $\alpha_1^{(i)} = 5.725$ cells/ms^2 (assuming $\lambda = log\ 2$). 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 average throughput in the equilibrium state is 336.7 cells/ms (i.e., $\overline{R}/\mu = 0.92$), which is higher than that of the rate-based scheme.

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

7.2 Bandwidth Guarantees and Fairness

In the second example, 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, $Q_h = 75$ 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, $\mu_1 = \frac{1}{3}\mu = 122.3$ cells/ms, $\mu_2 = \frac{2}{3}\mu = 244.7$ cells/ms, $Q_h^{(1)} = \frac{1}{3}Q_h = 25$ cells, $Q_h^{(2)} = \frac{2}{3}Q_h = 50$ cells, $Q_l^{(1)} = \frac{1}{3}Q_l = 7.5$ cells, $Q_l^{(2)} = \frac{2}{3}Q_l = 15$ cells, $\xi_1 = \frac{1}{3}\xi = 150$ cells, $\xi_2 = \frac{2}{3}\xi = 300$ cells. We take $\lambda = log\ 2$ in reducing rate increase parameter α.

We assume that VC1 starts sending data at $t = 0$ and VC2 starts sending data at $t = 195$ ms when VC1 has already reached the optimal equilibrium state. For the proposed scheme, using the expressions derived for Pattern I, II, and III in Sections 5 and 6, we compute the evolutions for $R_1(t)$ and $R_2(t)$ for both transient and equilibrium states as shown in Figure 16(b). 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 VC1, and thus, VC1 grabs all available bandwidth μ (ABR). Note that VC1's transient-cycle number $N^{(1)} = 2$ here, which validates the claim (1) of Theorem 2 since $N^{(1)}$'s exact value is $\lceil 1.6764 \rceil = 2$ by Eq. (A.4) with $\alpha_\xi = 7.1647$ and its upper-bound (attained in this case) is $\lceil 1.7734 \rceil = 2$ by Eq. (6.3). At $t = 195$ ms, VC2 starts sending data cells, then the WRR scheduler at the switch assigns a bandwidth share of μ_1 to VC1 and μ_2 to VC2 as their target bandwidths. VC2 starts competing for bandwidth in its transient cycles. At the same time, the equilibrium state of $R_1(t)$ is broken and starts to give up the bandwidth beyond its share μ_2. Note that VC1'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 and the system enters another equilibrium state. The claim (1) of Theorem 2 is also verified for VC2 since $N^{(2)} = 2$, and its exact value is $\lceil 1.4916 \rceil = 2$ by Eq. (A.4) with $\alpha_\xi = 16.2873$ and its upper-bound (attained) is $\lceil 1.7734 \rceil = 2$. 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, $[0, Q_{max}^{(1)}]$ and $[0, Q_{max}^{(2)}]$ where $Q_{max}^{(1)} = 131$ and $Q_{max}^{(2)} = 263$.

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-based scheme, as it does not explicitly control transmission rate.

These two examples also show that under the proposed scheme the source rate and queue length can rapidly converge to the optimal operating regime (within two cycles of the transient state).

8 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. Using the fluid approximation, we modeled the proposed flow-control scheme and analyzed the system dynamic behavior for ABR services under the most stringent traffic condition. We derived closed-form expressions for queue buildups, average throughput, and other flow-control measures. These expressions were then used to evaluate the system performance, design the optimal rate-control parameters, and compute the evolution of rate and queue-length functions.

Unlike other flow-control schemes and analyses, we included the buffer capacity as an important constraint in the design and analysis of the proposed scheme. Two important findings are: (1) the control pattern with no buffer congestion is optimal in term of average throughput and network delay, and (2) the rate-increase parameter α plays a critical role for the system to operate under the optimal pattern. Accordingly, we developed a 2-dimensional rate-control scheme which not only controls the increase/decrease of the rate ($R(t)$), but also adjusts its rate ($R'(t) = \frac{dR(t)}{dt}$). We derived a sufficient condition by which the second-order rate control can drive the system from any transient state to an optimal equilibrium state. Also, the worst-case performance bounds are derived as the closed-form functions of flow-control parameters to evaluate the performance of the second-order rate control in terms of convergence speed and buffer utilization. Our numerical results demonstrated that the worst-case performance bounds are very tight for the designated operating regime, and thus can be used to accurately estimate the performance of the second-order rate control. In addition, the worst-case bounds exhibited the tradeoff between the convergence speed and buffer-utilization/transient-response, which facilitates the selection of an appropriate second-order rate control parameter. The resulting evolutions of $R(t)$ and $Q(t)$ have shown the proposed flow-control scheme to be stable and efficient in that $R(t)$ and $Q(t)$ rapidly converge to the designated operating region.

Our numerical analyses indicate that, in equilibrium state, a smaller α results in a higher average throughput, but for quick transient response and high throughput, a large α is desirable. This observation is consistent with the conclusion in [5]: "a sharp reduction in the transient state and a smaller rate reduction in the equilibrium state". For a given α, our evaluation illustrated that in transient state a larger ξ leads to a higher throughput, but also results in a longer transient cycle, thus showing a tradeoff between throughput and transient cycle.

To make a quantitative performance comparison with existing schemes, we considered two multiple-connection examples. The identical VCs example showed that our scheme requires much smaller (nearly 1/5) buffer size to guarantee lossless transmission and achieves higher average throughput or bandwidth utilization (92% vs. 87%) than the rate-based scheme. The non-identical VCs example showed that our scheme can provide a bandwidth guarantee to each VC and achieve a fair bandwidth share among competing VCs according to their MCRs. Also shown in the examples is that $R(t)$ and $Q(t)$ can quickly converge to the optimal operation region.

We have developed a simulator using NetSim package [22]. The simulation results verified the analytical results for the single-connection case. We are currently extending the simulator to the multiple-connection case. Also to be simulated is the dynamic behavior of multiple connections using the realized bandwidth as the target bandwidth, which is more accurate, but analytically intractable. We are also planning to extend our model to the case where per-VC queue scheduling at a switch is implemented by more efficient scheduling algorithms such as Generalized Processor Sharing (GPS) and Fair Queueing (FQ) [23,25]. Effective link-scheduling algorithms, coupled with integrated rate- and credit-based flow control, can provide good end-to-end performance in emerging broadband networks.

Acknowledgment

The authors would like to thank Jennifer L. Rexford and Zhihui Huang for very helpful discussions and comments on an earlier version of this paper.

A Appendix

A.1 Proof of Theorem 2.

Proof. For claim (1), let α_ξ be the rate parameter corresponding to buffer capacity ξ such that $\xi = Q_{max}(\alpha_\xi)$. According to the definition of α^* (see Definition 1 in Section 6), if $\alpha_0 > \alpha_\xi$, then the number of transient-state cycles $N \geq 1$ and $\alpha^* = e^{-\lambda N}\alpha_0$, which can be rewritten as

$$N = \frac{1}{\lambda}\ log\ \left(\frac{\alpha_0}{\alpha^*}\right). \tag{A.1}$$

Since

$$\alpha^* \triangleq \max_{i\in\{1,2,3,\ldots\}}\{\alpha_i \mid Q_{max}^{(i)} \leq \xi\} = \max_{i\in\{1,2,3,\ldots\}}\{\alpha_i \mid \alpha_i = e^{-\lambda i}\alpha_0,\ Q_{max}(\alpha_i) \leq \xi\} \tag{A.2}$$

and thus

$$e^{-\lambda(N-1)}\alpha_0 > \alpha_\xi \geq \alpha^* = e^{-\lambda N}\alpha_0, \tag{A.3}$$

we obtain

$$N = \frac{1}{\lambda}\ log\ \left(\frac{\alpha_0}{\alpha^*}\right) = \left\lceil\frac{1}{\lambda}\ log\ \left(\frac{\alpha_0}{\alpha_\xi}\right)\right\rceil. \tag{A.4}$$

We now prove the following fact by contradiction

$$\sqrt{\alpha_\xi} \geq \left(\frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}\right). \tag{A.5}$$

Suppose that $\sqrt{\alpha_\xi} < \left(\frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}\right)$, then $\exists\ \alpha'$, a positive real number, such that

$$\sqrt{\alpha_\xi} < \sqrt{\alpha'} < \left(\frac{\sqrt{\xi}-\sqrt{2Q_h}}{\tau}\right). \tag{A.6}$$

Applying claim (1) of Theorem 1 to the second part of Eq. (A.6), we get $Q_{max}(\alpha') \leq \xi$, but the first part of Eq. (A.6) implies $Q_{max}(\alpha') > \xi$ since $Q_{max}(\alpha)$ is a strictly monotonic-increasing function of α for $\tau > 0$, which creates a contradiction. Hence, Eq. (A.5) holds true. Substituting Eq. (A.5) into Eq. (A.4), Eq. (6.3) follows.

For claim (2), by definition of α^*, which is a real function of α_0, we have

$$\alpha_\xi \geq \alpha^* = \alpha^*(\alpha_0) > e^{-\lambda}\alpha_\xi \qquad \forall\alpha_0 > \alpha_\xi \tag{A.7}$$

Since the value of real function $\alpha^*(\alpha_0)$ can be arbitrarily close to $e^{-\lambda}\alpha_\xi$, but never attains it, due to the arbitrariness of α_0, we get

$$\inf_{\forall \alpha_0 > \alpha_\xi} \{\alpha^*(\alpha_0)\} = e^{-\lambda}\alpha_\xi \tag{A.8}$$

Applying the fact described by Eq. (A.5) to Eq. (A.8), Eq. (6.4) follows. □

References

[1] H. T. Kung and A. Chapman, "The FCVC (Flow-Controlled Virtual Channels) proposed for ATM networks," in *Proc. International Conf. on Network Protocols*, pp. 116–127, 1993.

[2] H. T. Kung and A. Chapman, "Use of link-by-link flow control in maximizing ATM network performance: simulation results," in *Proc. IEEE Hot Interconnects Symp.*, 1993.

[3] 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.

[4] H. T. Kung and K. Chang, "Receiver-oriented adaptive buffer allocation in credit-based flow control for ATM networks," in *Proc. of IEEE INFOCOM*, pp. 239–252, April 1995.

[5] 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.

[6] L. Roberts, *Enhanced Proportional Rate-Control Algorithm*, ATM Forum contribution 94-0735R1, September 1994.

[7] H. Hsiaw et al., *Closed-loop Rate-based Traffic Management*, ATM Forum contribution 94-0438R2, September 1994.

[8] L. Roberts and A. Barnhart, *New Pseudocode for Explicit Rate Plus EFCI Support*, ATM Forum contribution 94-0974, October 1994.

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

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

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

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

[13] 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.

[14] 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.

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

[16] G. Newell, *Applications of Queueing Theory*, Chapman and Hall, 1971.

[17] L. Kleinrock, *Queueing systems - Volume II, Computer Applications*, John Wiley and Sons, 1976.

[18] 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.

[19] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Analysis of rate-based congestion control algorithms for ATM networks—Part 2: initial transient state analysis—," in *Proc. of GLOBECOM*, pp. 1095–1101, November 1995.

[20] 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.

[21] X. Zhang, K. G. Shin, D. Saha, and D. Kandlur, "Scalable flow control for multicast ABR services in ATM networks," *submitted for publication*, July 1997.

[22] A. Heybey, *The Network Simulator*, Laboratory for Computer Science, Massachusetts Institute of Technology, October 1989.

[23] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[24] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Trans. on Networking*, vol. 2, no. 2, pp. 137–150, April 1994.

[25] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," in *Proc. of ACM SIGCOMM*, pp. 1–12, 1989.