Scalable Flow Control for Multicast ABR Services in ATM Networks

Xi Zhang† Kang G. Shin† Debanjan Saha‡ Dilip Kandlur‡

†Department of EECS
The University of Michigan
Ann Arbor, MI 48109
{*xizhang,kgshin*}*@eecs.umich.edu*

‡Network Systems Department
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{*debanjan,kandlur*}*@watson.ibm.com*

ABSTRACT

We propose an efficient flow-control scheme for ATM ABR multicast services. We develop a second-order rate-control algorithm to deal with the variation in feedback delay resulting from dynamic "drift" of the bottleneck location within a multicast tree. The proposed scheme makes the rate process converge to the available bandwidth of the multicast-connection's most congested link. It also confines the buffer occupancy to a target regime bounded by a given (finite) buffer capacity at the bottleneck node. Using fluid approximation, we model the proposed scheme and study the system dynamics under the most stressful traffic conditions. We derive expressions for queue buildups and average throughputs in both transient and equilibrium states. We identify the system control factors that govern the system dynamics and develop an optimal control condition which guarantees monotonic convergence of the system state to the target regime from an arbitrary initial value.

Index Terms — Multicast flow control, scalable algorithms, ATM networks, ABR service, best-effort service, closed-loop feedback control, rate-based flow control, lossless transmission, fairness.

The work reported in this paper was supported in part by the ONR under Grant N00014-94-0229. 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 agency.

1 Introduction

The Available-Bit-Rate (ABR) service, targeted primarily towards adaptive applications, is an increasingly popular and important class of service in ATM networks. Ever since the conception of ABR service by the ATM Forum in September 1994, it has attracted significant attention from researchers in the networking community. While the literature on ABR is extremely rich, a vast majority of it focus only on point-to-point (unicast) connections. Our objective in this paper is to develop efficient and scalable control mechanisms for supporting ABR service on point-to-multipoint (pt-to-mpt or multicast) connections.

Supporting ABR pt-to-mpt connection service poses a number of new challenges not encountered in providing ABR service on unicast connections. One of the major problems, especially in large multicast trees, is commonly known as the feedback implosion problem [1]. Since our goal is to adjust the source transmission rate to match the bottleneck link bandwidth, the source need to collect congestion feedback from all branches in the multicast tree. Simultaneous congestion feedback from all branches can cause an implosion at the source, especially when the multicast tree is large. Hence, for reasons of scalability, it is important to consolidate the congestion feedback at each branch point and only the consolidated feedback is forwarded upstream. Consolidation requires synchronization of feedback from all downstream branches of each branch point. Since different branches may have different round trip delays, receiver-generated feedback may arrive at the branch point at significantly different times. If a branch-point switch waits for feedback from all of its downstream nodes, it may have to wait a long time, thus resulting in a long feedback delay. On the other hand, if the branch-point switch forwards an early feedback upstream without waiting for feedback from all of its downstream nodes, the source may receive incomplete/incorrect information.

Another important but subtle problem in multicast flow control is that the bottleneck may shift from one path to another. As a result, the round trip delay in the bottleneck path may change significantly. Since the round trip delay plays a critical role in determining the effectiveness of any feedback flow-control scheme, it is important to identify and handle such dynamic drifts of the bottleneck. The flow-control scheme should also be able to detect and remove non-responsive branches in order to prevent them from stalling the entire connection.

Roberts [2,3] proposed a multicast flow-control scheme which is based on EPRCA (Explicit Proportional Rate Control Algorithm) and extends unicast to multicast operations. The authors of [4–6] established a framework for extending an existing unicast congestion-control protocol to a multicast environment. The features of these two schemes are simple, and easy to migrate from a unicast environment to a multicast environment. They both employ a simple level-by-level feedback mechanism where feedback RM cells from downstream nodes are aggregated and then sent upward whenever a forward RM cell cell is received at each level of the multicast tree. While level-by-level feedback is conceptually very simple, it suffers from the problems of large feedback delays, feedback inefficiency (due to no synchronization), and poor scalability (round trip feedback delay proportional to the height of the multicast tree).

In contrast to the schemes presented in [2–6], we propose a new scheme that employs leaf-to-root feedback that achieves excellent feedback efficiency. At the heart of our solution is a second-order rate-control algorithm. More specifically, besides adapting the transmission rate based on the congestion feedback, the source also adjusts the second-order parameters that determine the rate at which the transmission rate itself is adjusted. We show that this second-order rate-control mechanism helps the source adapt itself to the changes in round trip delay based on leaf-to-root feedback even when the bottleneck location drifts from one path to another. We use a soft synchronization protocol for consolidation of feedbacks at each branch-point. This not only solves the feedback implosion and synchronization problems, but also makes the round trip delay independent of the height and structure of the multicast tree, and readily detects and eliminates non-responsive branches.

Using the fluid approximation, we model the proposed scheme and develop an optimal control condition, under which the second-order rate-control guarantees the monotonic convergence of system state to the optimal regime from an arbitrary initial value. We analytically derive the relationship between the rate-control parameter and round trip delay subject to finite buffer-capacity constraints. We derive expressions for queue buildups and average throughputs in both transient and equilibrium states as functions of rate-control parameters, feedback delay, target bandwidth, and target buffer occupancy. The results show that the proposed scheme is efficient and stable in the sense that both the source rate and queue length at the bottleneck rapidly converge to a small neighborhood of the designated operating point.

The paper is organized as follows. Section 2 describes the proposed scheme in detail. Section 3 introduces the concept of system bottleneck and leaf-to-root feedback delay, and constructs the system and control models for the proposed scheme. Section 4 describes the second-order rate-control law and demonstrates how it adapts itself to round trip delays so as to achieve lossless transmission. In Section 5, we derive analytical solutions for both transient and equilibrium states and evaluate the scheme's performance for the single-connection case. Section 6 deals with modeling and performance analysis for the cases of multiple concurrent multicast connections. The paper concludes with Section 7.

2 The Proposed Scheme

As in the existing schemes, we also use the EFCI (Explicit Forward Congestion Indication) bit and RM (Resource Management) cells to convey network congestion information. However, we refine the RM cell format [7] such that it contains both the cell-rate (first-order) control and the rate-parameter (second-order) control information. More specifically, two new one-bit fields, BCI (Buffer Congestion Indication) and NMQ (New Maximum Queue), are defined. Our scheme classifies congestion into two types: (1) when the queue length $Q(t)$ at a switch exceeds a predetermined threshold Q_h, we call it *bandwidth congestion*. Under this condition the switch sets the local CI (Congestion Indication) bit. (2) when the maximum queue length Q_{max} at a switch exceeds the target buffer occupancy $Q_{goal}, (Q_h < Q_{goal} < C_{max})$, where C_{max} is the buffer capacity, we call it *buffer congestion*. Under this condition the switch sets the local BCI state to 1.

Switches maintain a congestion state for each pt-to-mpt connection passing through them. When a switch receives a backward RM cell from a downstream node, it consolidates the explicit rate (ER), congestion indication (CI), and buffer congestion indication (BCI) for the associated connection. The ER is set to the minimum of the ER computed by the branch-point node and the ER received from the downstream nodes of the multicast tree. The CI field associated with the connection state is set to 1, if the local CI state is 1, or CI field in RM cell received form any one of the downstream neighbors is 1. The BCI field associated with the connection state is computed in the same way. When feedback from all downstream nodes have been accumulated, a single feedback RM cell is generated with the consolidated congestion information and sent upward with CI and BCI fields set to their respective values in the connection state. Note that our algorithm allows branch-point switches to consolidate feedback information from the backward RM cells that are generated by the leaves in response to *different* forward RM cells. This distinguishes our algorithm from both (i) "strict synchronization", where only the backward RM cells generated in response to the same forward RM cell are consolidated, thus making feedback delay determined by the longest branch, and (ii) "no synchronization" at all in [2–6], which may produce incomplete feedback information to the source and thus defeats the feedback efficiency. In the proposed scheme, the feedback RM cells are "softly synchronized" at each branch-point, providing fast, complete, and efficient feedback, and making the feedback delay scalable with the height and structure of multicast tree. Additionally, each node also dynamically identifies the non-responsive downstream nodes and removes them from the set of state variables associated with the responsive branches. For lack of space, we do not describe this part of algorithm in detail.

There are two rate-control modes at the source corresponding to the two types of congestion respectively: (1) bandwidth congestion control, and (2) buffer congestion control. If the bandwidth congestion information with CI = 1 (or 0) is detected from a feedback RM cell, then the cell rate is reduced multiplicatively (or increased additively) from its current value. The buffer congestion control is triggered when the source detects a transitions from CI = 1 to CI = 0 (i.e., from a rate-decrease cycle to a rate-increase cycle). Depending on the state of the BCI field, three different variations of this control is exercised by the source. If the BCI indicators in both the current and the last RM cells received are set to 0s, the rate-increase parameter is increased additively. When BCI indicator toggles from 1 to 0, the rate-increase parameter is increased multiplicatively. If the current RM cell has BCI field set to 1, the rate-increase parameter is decreased multiplicatively. Each time when buffer congestion control is triggered, the source sets NMQ field to 1 in the next forward RM cell to "request" the switches to recalculate Q_{max} for the next measurement-cycle.

A snippet of the pseudocode for the source control algorithm is presented in Figure 1. This algorithm deals with the receipt of feedback RM cells. Upon receiving a feedback RM cell, the source must first check if it is the time to exercise the buffer-congestion (second order) control. This algorithm is triggered when the source detects a transition from a rate decrease cycle to a rate increase cycle, that is when LCI (local congestion indicator) 1, and the CI field in the RM cell received is set 0. In this phase, the rate-increase parameter is adjusted depending on the current state of the local BCI indicator (LBCI) and the state of the BCI field (BCI) in the RM cell received. As mentioned before, we consider three cases: (i) if BCI is set to 1 in the RM cell received, the rate increase parameter AIR (Additive Increase Rate) is decreased multiplicatively by a factor of q ($0 < q < 1$), (ii) if both LBCI and BCI are

```
On receipt of an RM cell:
    if (CI=0 ∧ LCI=1)                          ! Buffer congestion control
        begincase                              ! Consider different cases
            case (BCI=1):                      ! BCI is set in the RM cell
                  $AIR \leftarrow q \times AIR$          ! AIR (Additive Increase Rate)
            case (BCI=0 ∧ LBCI=0):             ! BCI stays at 0
                  $AIR \leftarrow p + AIR$
            case (BCI=0 ∧ LBCI=1):             ! BCI toggles from 1 to 0
                  $AIR \leftarrow AIR/q$
        endcase
        $MDF \leftarrow e^{-AIR/BW\_EST}$              ! MDF (Multiplicative Decrease Factor)
        $LNMQ = 1$                             ! Start a new measurement cycle
    endif

    if (CI=0)                                  ! Bandwidth control
        $ACR = ACR + AIR$                      ! Increase cell rate additively
    else
        $ACR = ACR \times MDF$                 ! Decrease cell rate multiplicatively
    endif

    $LCI \leftarrow CI$                          ! Save the CI value
    $LBCI \leftarrow BCI$                        ! Save the BCI value
```

Figure 1: Source control algorithm.

set to 0, the rate-increase parameter AIR is increased additively by a step-size of p ($p > 0$), (iii) if LBCI is 1 and BCI is 0, AIR is increased multiplicatively by the same factor of q. For all these 3 cases, the rate-decrease parameter MDF (Multiplicative Decrease Factor) is adjusted according to the estimated bottleneck bandwidth *BW_EST* (see Section 4.2 for a detailed account). Additionally, the local new maximum queue indication (NMQ) bit is marked and BCI field is saved in LBCI . The source always exercises the cell-rate (first order) control whenever an RM cell is received. Using the same, or updated, rate-parameter, the source additively increases, or multiplicatively decreases, its ACR (Allowed Cell Rate) according to the CI field in received RM cell. The new CI is saved in LCI for the second-order rate control.

At the receiver, when a data cell is received, the EFCI bit is saved. When an RM cell is received, its CI bit is set using the EFCI bit saved from the data cell last received. The RM cell is then sent backward with the congestion information.

3 The System Model

An ATM network with ABR pt-to-mpt connections flow-controlled by the proposed scheme is a dynamic feedback-control system. We model this system using the first-order fluid approximation, which models the system with coupled time-delayed differential equations [8,9]. In this model, we use the

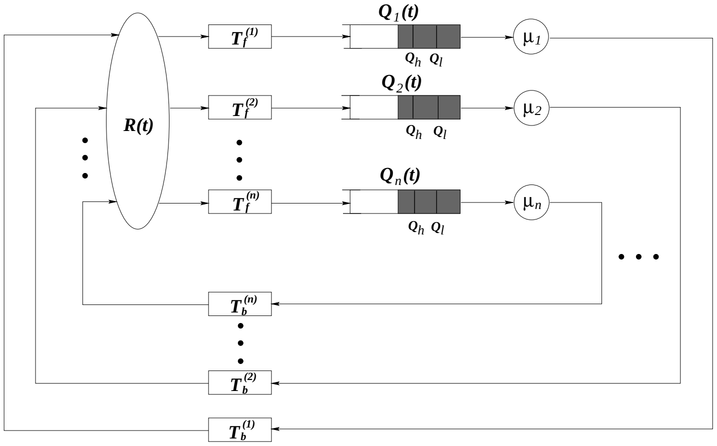


Figure 2: The system model for a pt-to-mpt connection.

real-valued deterministic functions $R(t)$ and $Q(t)$ to approximate a discrete 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 has been used effectively for the analysis and evaluation of ABR unicast flow-control schemes [9–17].

The existence of multiple paths in an ABR pt-to-mpt connection complicates its modeling and analysis. As in any feedback-control system, the RM cell round trip delay plays a critical role in determining system performance. In all previous analyses of unicast flow control using the fluid model, the round trip delay is treated as a *constant* equal to the value found/determined during the setup of each ABR connection [9–17]. However, as mentioned earlier, the RM cell round trip delay in a pt-to-mpt connection varies significantly with time. Our model, therefore, takes this variation into account. By applying the proposed second-order rate control (to be discussed in Section 4), the source rate-control can be adapted quickly to the variation of RM cell round trip delay so as to guarantee lossless transmission for a given buffer size. We also assume the existence of only a single bottleneck[1] at a time with queue length $Q(t)$ and a "persistent" source with ACR $= R(t)$ for each pt-to-mpt connection. Such a data-source model enables us to examine the proposed scheme under the most stressful condition. Figure 2 depicts the system model for a pt-to-mpt connection flow-controlled by the proposed scheme.

3.1 System Description

As shown in Figure 2, a pt-to-mpt connection consists of n paths with RM cell round trip delays $\tau_1, \tau_2, \cdots, \tau_n$, and bottleneck bandwidths $\mu_1, \mu_2, \cdots, \mu_n$. There is only a single bottleneck on each path and its location may change with time. Thus, we use $T_f^{(i)}$ to represent the "forward" delay from the source to the bottleneck, and $T_b^{(i)}$ the "backward" delay from the bottleneck to the source via the destination node of the i-th path. Clearly, $T_b^{(i)} = \tau_i - T_f^{(i)}$. Each path's bottleneck has its

[1] This is not a restriction, because the bottleneck is defined as the most congested link/switch.

own $Q_i(t), i = 1, 2, \cdots, n$. According to the proposed control algorithm, all paths of a flow-controlled pt-to-mpt connection share the same $R(t)$ which dictates every path's dynamic behavior. As a result, all the paths in a pt-to-mpt connection "interact" with each other via their "shared" $R(t)$. Thus, the system model consists of n coupled subsystems, each corresponding to an individual path of the pt-to-mpt connection. The i-th subsystem/path is characterized by the following parameters:

β:	Multiplicative decrease factor for rate reduction
α:	Additive rate-increase slope
Δ:	Rate-update time interval
$Q_h^{(i)}$ $(Q_l^{(i)})$:	High (low) threshold of the ABR queue
$T_b^{(i)}$ $(T_f^{(i)})$:	Backward (forward) delay
ξ_i:	Bottleneck's maximum buffer allocation (C_{max})
μ_i:	Bottleneck link bandwidth (BW)

We use the synchronous model for rate control in which the periodic update interval Δ is usually a fraction of the round trip delay. $Q_h^{(i)}$ and $Q_l^{(i)}$ are used for detecting traffic overload and underload, respectively. Based on the proposed control algorithms in Section 2, the additive increase and the multiplicative decrease of rate during the n-th rate-update interval are expressed as:

$$R_n = \begin{cases} R_{n-1} + a; & \text{additively increase } (a > 0) \\ bR_{n-1}; & \text{multiplicatively decrease } (0 < b < 1) \end{cases} \tag{3.1}$$

where a (AIR in the source node algorithm) is the rate increment, and b (MDF in the source node algorithm) the rate decrease factor. Thus, the rate adjustment at the source can be modeled by linear-increase and exponential-decrease in a continuous-time domain as follows [10]:

$$R(t) = \begin{cases} R(t_0) + \alpha(t - t_0); & \text{linear increase } (\alpha > 0) \\ R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}}; & \text{exponential decrease } (\beta < 1) \end{cases} \tag{3.2}$$

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

3.2 System Control Factors

At any given time, the most congested path of a pt-to-mpt connection governs the dynamic behavior of the flow-control system. To explicitly model this feature, we introduce the following definition.

Definition 1 *The* ***system bottleneck*** *is the bottlenecked path whose feedback dictates the source rate-control actions. The* ***system feedback bottleneck delay*** *is the* RM *cell round trip delay experienced on the system bottleneck.*

According to the proposed algorithm, the rate-control actions are based on the following feedback signals: (1) $ER = \min_{i\in\{1,2,\cdots,n\}}\{ER(i)\}$; (2) $CI = \bigcup_{i\in\{1,2,\cdots,n\}}\{CI(i)\}$; (3) $BCI = \bigcup_{i\in\{1,2,\cdots,n\}}\{BCI(i)\}$ where $ER(i)$, $CI(i)$, and $BCI(i)$ are the requested minimum rate, bandwidth-congestion indication, and buffer-congestion indication for path i, respectively. Obviously, ER is determined by the minimum bottleneck bandwidth, and CI or BCI is marked first in the path with minimum available bandwidth. Thus, the system bottleneck is located along the path which has the minimum bottleneck bandwidth. Since the system bottleneck dictates the source rate-control actions, we can analyze the multicast flow-control system by focusing on its system bottleneck's state equations. Let $Q(t)$ be the queue length function at the system bottleneck and $\tau = T_f + T_b$ be the system bottleneck feedback delay. Then, the system bottleneck state is specified by the two state variables, $R(t)$ and $Q(t)$. According to the proposed algorithms, the system bottleneck state equations are given as:

Source-rate function:

$$R(t) = \begin{cases} R(t_0) + \alpha(t - t_0); & \text{if } Q(t - T_b) < Q_h \text{, condition for the rate-control rule} \\ & \text{to switch to linear increase} \\ R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}}; & \text{if } Q(t - T_b) \geq Q_h \text{, condition for the rate-control rule} \\ & \text{to switch to exponential decrease} \end{cases} \tag{3.3}$$

System bottleneck queue-length functions:

$$Q(t) = \begin{cases} 0; & \text{if } Q(t) = 0 \text{ and } R(t) < \mu \\ \int_{t_0}^{t} [R(v - T_f) - \mu] dv + Q(t_0); & \text{if } R(t) > \mu \text{, or if } R(t) < \mu \text{ and } Q(t) > 0 \end{cases} \tag{3.4}$$

where $\mu = \min\{\mu_1, \mu_2, \cdots, \mu_n\}$ is the system bottleneck bandwidth, and Q_h is the high queue-threshold for the system bottleneck's buffer. Here $R(t)$ represents the fluid approximation to cell-transmission throughput. 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 system bottleneck queue-length process. This first-order fluid model has been shown in [8] to be a good approximation when the system is heavily-loaded.

As mentioned earlier, the system bottleneck dynamically drifts around different paths of the pt-to-mpt connection as the cross-traffic of other connections varies with time. Consequently, the system bottleneck feedback delay also dynamically changes with time, which can significantly affect the performance of multicast flow-control. Thus, we explicitly include the variation of round trip delay in our model, and study its relationship with multicast flow-control algorithms. We discuss this issue in the next section.

4 The Second-Order Rate-Control

As discussed in [17], increasing or decreasing $R(t)$ is not effective enough to have Q_{max} upper bounded by C_{max} when the system bottleneck feedback delay τ varies. This is because rate-increase/decrease control can only make $R(t)$ fluctuate around the designated bandwidth, but cannot adjust the rate-fluctuation amplitude that determines Q_{max}. It can be shown that Q_{max} increases with both τ and rate-increase parameter $\alpha = \frac{dR(t)}{dt}$ and can be written as, $Q_{max}(\tau, \alpha)$, or $Q_{max}(\alpha)$ for a given τ. Thus Q_{max} can be controlled by adjusting α in response to the variation of τ. The control over α — which we call *α-control* — is the second-order control over $R(t)$, providing one more dimension to control the dynamics of the proposed flow-control.

4.1 α-Control

The α-control is a discrete-time control process since it is only exercised when the source rate control is in a "decrease-to-increase" transition based on the the buffer congestion feedback signal, $BCI(n) = 0$ (1) if $Q_{max}^{(n)} \leq Q_{goal}$ ($Q_{max}^{(n)} > Q_{goal}$), where Q_{goal} ($Q_h < Q_{goal} < C_{max}$) is the target buffer occupancy (also called *setpoint*) in the equilibrium state. If the system bottleneck shifts from a shorter path to a longer one, then τ will increase, making Q_{max} larger. When Q_{max} eventually grows beyond Q_{goal}, buffer will overflow, implying that the current α is too large for the increased τ. The source must reduce α to prevent cell loss. On the other hand, if τ decreases from its current value due to the shift of the system bottleneck from a longer path to a shorter one, then Q_{max} will decrease. When $Q_{max} < Q_{goal}$, only a small portion of buffer space will be utilized, implying that the current α is too small for the decreased τ. The source should increase α to avoid buffer under-utilization and to improve the system responsiveness in grabbing available bandwidth. Keeping $Q_h < Q_{goal} < C_{max}$ has two benefits: (1) the source can quickly grab available bandwidth; (2) it can achieve high throughput and high bandwidth utilization.

The main purpose of α-control is to handle the buffer congestion resulting from the variation of τ. We set three goals for α-control: (1) ensure that $Q_{max}^{(n)}$ quickly converges to, and stays within, the neighborhood of Q_{goal}, which is upper-bounded by C_{max}, from an arbitrary initial value by driving their corresponding rate-increase parameters α_n to the neighborhood of α_{goal} for a given τ; (2) maintain statistical fairness on the buffer occupancy among multiple pt-to-mpt connections which share a common system bottleneck; (3) minimize the extra cost incurred by the α-control algorithm. To achieve these goals, we propose a "converge and stay" α-control law in which the new value α_{n+1} is determined by α_n, and the feedback information BCI on Q_{max}'s current and one-step-old values, $Q_{max}^{(n)}$ and $Q_{max}^{(n-1)}$. The α-control law can be expressed by the following equations:

$$\alpha_{n+1} = \begin{cases} \alpha_n + p; & \text{if } BCI(n-1,n) = (0,0) \quad (Q_{max}^{(n-1)} \leq Q_{goal} \wedge Q_{max}^{(n)} \leq Q_{goal}) \\ q\alpha_n; & \text{if } BCI(n-1,n) = (0,1) \quad (Q_{max}^{(n-1)} \leq Q_{goal} \wedge Q_{max}^{(n)} > Q_{goal}) \\ q\alpha_n; & \text{if } BCI(n-1,n) = (1,1) \quad (Q_{max}^{(n-1)} > Q_{goal} \wedge Q_{max}^{(n)} > Q_{goal}) \\ \alpha_n/q; & \text{if } BCI(n-1,n) = (1,0) \quad (Q_{max}^{(n-1)} > Q_{goal} \wedge Q_{max}^{(n)} \leq Q_{goal}) \end{cases} \tag{4.1}$$

where q is the α-decrease factor such that $0 < q < 1$ and p is the α-increase step-size whose value will be discussed next.

4.2 The Properties of the α-Control

To characterize the α-control convergence, we first introduce the following two definitions.

Definition 2 *The* ***neighborhood*** *of target buffer occupancy Q_{goal} is specified by $\{Q_{goal}^l, Q_{goal}^h\}$ with*

$$Q_{goal}^l \triangleq \max_{n \in \{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \leq Q_{goal}\} \tag{4.2}$$

$$Q_{goal}^h \triangleq \min_{n \in \{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \geq Q_{goal}\} \tag{4.3}$$

where $Q_{max}^{(n)}$ is governed by the proposed α-control law.

Definition 3 $\{Q_{max}^{(n)}\} \triangleq \{Q_{max}(\alpha_n)\}$ *is said to* ***monotonically*** *converge to Q_{goal}'s neighborhood at time $n = n^*$ from its initial value $Q_{max}^{(0)} = Q_{max}(\alpha_0)$, if $BCI(0,1,2,3,\cdots,n^*-1,n^*,n^*+1,n^*+2,n^*+3,\cdots) = (0,0,0,0,\cdots,0,1,0,1,0,\cdots)$ for $\alpha_0 < \alpha_{goal}$; and $BCI(0,1,2,3,\cdots,n^*-1,n^*,n^*+1,n^*+2,n^*+3,\cdots) = (1,1,1,1,\cdots,1,0,1,0,1,\cdots)$ for $\alpha_0 > \alpha_{goal}$.*

The α-control is applied either in a *transient* state, during which $Q_{max}^{(n)}$ has not yet reached Q_{goal}'s neighborhood, or in an *equilibrium* state, in which $Q_{max}^{(n)}$ fluctuates within Q_{goal}'s neighborhood periodically. The α-control aims at making $Q_{max}^{(n)}$ converge fast in transient state and staying steadily within its neighborhood in equilibrium state. The following theorem summarizes the α-control law's convergence properties, operating conditions, and the method of computing the control parameter in both the transient and equilibrium states. Note that Q_{goal}^{l} and Q_{goal}^{h} are the closest attainable points around Q_{goal}, but Q_{goal} may not necessarily be the midpoint between Q_{goal}^{l} and Q_{goal}^{h}. The actual location of Q_{goal} between Q_{goal}^{l} and Q_{goal}^{h} depends on all rate control parameters and initial value of α_0.

Theorem 1. *Consider the proposed α-control law Eq. (4.1) which is applied to a pt-to-mpt connection with its system bottleneck characterized by Q_{goal}, Q_h, and τ.* ***If*** *(1) $\alpha = \alpha_0$, an arbitrary initial value at time n, (2) $0 < q < 1$, and (3) $p \le \left(\frac{1-q}{q}\right)\left(\frac{\sqrt{Q_{goal}}-\sqrt{2Q_h}}{\tau}\right)^2$,* ***then*** *(1) in the transient state the α-control law guarantees $Q_{max}^{(n)}$ to* ***monotonically*** *converge to Q_{goal}'s neighborhood, and (2) in the equilibrium state the fluctuation amplitudes of $Q_{max}^{(n)}$ around Q_{goal} are bounded as follows:*

$$\begin{cases} Q_{goal}^{h} - Q_{goal} \le \tau^2\alpha_{goal}\left(\frac{1}{q}-1\right) + \tau\sqrt{8\alpha_{goal}Q_h}\left(\frac{1}{\sqrt{q}}-1\right), \\ Q_{gaol} - Q_{goal}^{l} \le \tau^2\alpha_{goal}(1-q) + \tau\sqrt{8\alpha_{goal}Q_h}(1-\sqrt{q}), \end{cases} \tag{4.4}$$

and the diameter of neighborhood for the target buffer occupancy Q_{goal} is bounded by

$$Q_{gaol}^{h} - Q_{goal}^{l} \le \tau^2\alpha_{goal}\left(\frac{1}{q}-q\right) + \tau\sqrt{8\alpha_{goal}Q_h}\left(\frac{1}{\sqrt{q}}-\sqrt{q}\right), \tag{4.5}$$

where α_{goal} is the rate-increase parameter corresponds to Q_{goal}.

Proof: The proof is omitted here due to lack of space. □

Remarks: The α-control law is similar to, but differs from, additive-increase/multiplicative-decrease algorithm in the following terms. During the transient state, the α-control law behaves like an additive-increase/multiplicative-decrease algorithm, which accommodates statistical convergence-to-fairness of buffer utilization among the multiple pt-to-mpt connections sharing a common system bottleneck. On the other hand, in the equilibrium state, the α-control law guarantees the buffer occupancy to be locked with its setpoint the first time when $Q_{max}^{(n)}$ reaches Q_{goal}'s neighborhood, regardless of the initial value α_0. In contrast, the additive-increase/multiplicative-decrease does not guarantee this monotonic convergence since α-control is a time-discrete control process and its convergence is α_0-dependent. The monotonic convergence ensures that $Q^{(n)}_{max}$ quickly converges to, and stays within, the neighborhood of its target value Q_{goal}. The extra cost paid for achieving these benefits is minimized since only a binary bit, BCI , is conveyed from the system bottleneck and two bits are used to store the current and one-step-old feedback information, $BCI(n-1)$ and $BCI(n)$, at the source. The α-increase step-size p specified in condition (3) in Theorem 1 is a function of α-decrease factor q. A large q (small decrease step-size) requests a small p for the monotonic convergence. By the condition (3) if $q \rightarrow 1$, then $p \rightarrow 0$, which is expected since for a steady convergency system, zero decrease corresponds to zero increase in system state. According to Eq. (4.4) and Eq. (4.5), when $q \rightarrow 1$, $Q^l_{goal}, Q^h_{goal} \rightarrow Q_{goal}$, i.e., $Q^{(n)}_{max}$'s fluctuation amplitude approaches to zero, which also makes sense since $q \rightarrow 1$ implies $p \rightarrow 0$, thus $Q^{(n)}_{max}$ approaches to a constant for all n.

To balance $R(t)$'s increase and decrease rates and to ensure the average of the offered traffic load not to grow beyond the bottleneck bandwidth, each time when α_n is updated by the α-control law specified by Eq. (4.1), the proposed algorithm also updates β_n as

$$\beta_n = 1 - \frac{\alpha_n}{\mu}\Delta. \tag{4.6}$$

Since α_n represents $R(t)$'s increase rate and $(1-\beta_n)/\Delta$ determines $R(t)$'s decrease speed, by rewriting Eq. (4.6) we get $\alpha_n / \left(\frac{1-\beta_n}{\Delta}\right) = \mu$, and hence, near the point $(R(t) = \mu,\ Q(t) = 0)$, the rate of $R(t)$'s increase is equal to its decrease rate. In addition, letting $\alpha_n \left(\frac{\Delta}{1-\beta_n}\right) = \mu$ reduces to a simpler scenario where we only need to control one parameter α_n, instead of both α_n and β_n.

5 System-Bottleneck Dynamics of a Single Connection

We derive analytical expressions for both equilibrium and transient state dynamics, which determine such performance measures as maximum queue length, average throughput, and oscillation periods of the rate/queue-length function. Also derived are expressions which can be used to compute the evolution of rate/queue-length functions.

5.1 Equilibrium-State Analysis

The system is said to be in the equilibrium state if $R(t)$ and $Q(t)$ have already converged to a certain regime and oscillate with a constant frequency and a steady average amplitude. In this state, $R(t)$ fluctuates around μ, and $Q^{(n)}_{max}$ around Q_{goal}. The fluctuation amplitudes and periods are determined by the rate-control parameters α, β; link bandwidth μ; target buffer occupancy Q_{goal}; α-control parameters p, q; congestion detection thresholds Q_h, Q_l; and delays T_b, T_f. To simplify the analysis of equilibrium state, we assume that the α-control parameters (i.e., α_0, Q_{goal}, p, and q) are properly selected according to the conditions specified in Theorem 1, such that $Q^{(n)}_{max}$ converges to a symmetric neighborhood of Q_{goal} where $Q_{goal} = \frac{1}{2}(Q^l_{goal} + Q^h_{goal})$ and $Q^h_{goal} < C_{max}$.

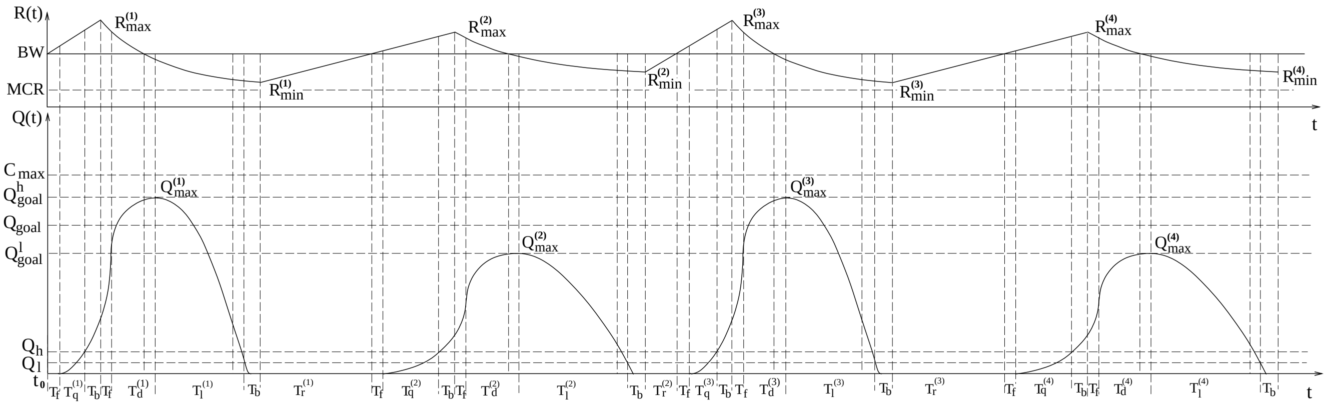


Figure 3: Dynamic behavior of $R(t)$ and $Q(t)$ for a single pt-to-mpt connection.

Figure 3 illustrates the first 4 cycles of rate fluctuation and the associated queue-length function at the bottleneck link in the equilibrium state with $\alpha_1 = \alpha_{goal}^{h}$. At time t_0, the rate reaches the link bandwidth μ and the queue starts to build up after a delay of T_f. At time $t_0 + T_b + T_q^{(1)}$, $Q(t)$ reaches Q_h and bandwidth congestion is detected. After a backward delay of T_b, the source receives CI $= 1$ feedback and its rate begins to decrease exponentially. $Q(t)$ reaches the peak as $R(t)$ drops back to the link bandwidth μ. When the rate falls below the link bandwidth, $Q(t)$ starts to decrease. After a time period of T_l elapsed, $Q(t)$ reaches Q_l, then the non-congestion condition (CI $= 0$) is detected and sent backward to the source. After a backward delay of T_b, the (CI $= 0$) feedback arrives at the source, then the rate-decrease to rate-increase transition condition is detected at the source. Subsequently, the source updates the next rate-increase parameter α_2 with a smaller value of $q\alpha_1$ (β_2 is also updated accordingly using Eq. (4.6)) since BCI $= 1$ (due to $Q_{max}^{(1)} > Q_{goal}$) is received in the feedback RM cell. Then, the source rate increases linearly with the new rate-parameter $\alpha_2 = q\alpha_1 = \alpha_{goal}^{l}$. When $R(t)$ reaches μ after a time period of $T_r^{(1)}$, the system starts the second fluctuation cycle.

The dynamic behavior of the second cycle of fluctuation follows a similar pattern to that in the first cycle except for the updated rate-control parameters α_2 and β_2 resulting in a longer cycle length due to smaller increase/decrease rates. When the transition from rate-decrease to rate-increase is detected again for the second fluctuation cycle, the source sets $\alpha_3 = \alpha_2/q$ because $Q_{max}^{(2)} < Q_{goal}$ i.e., $BCI(2) = 0$, hence $BCI(1,2) = (1,0)$. But $\alpha_3 = \alpha_2/q = (q\alpha_1)/q = \alpha_1$ since α_n has already converged to $\{\alpha_{goal}^{l}, \alpha_{goal}^{h}\}$ in the equilibrium state. Thus, the dynamic behavior of the third fluctuation cycle is exactly the same as the first cycle. In general, all odd-numbered fluctuation cycles have the same dynamic pattern and all even-numbered fluctuation cycles have another identical pattern, i.e., $\alpha_{2i} = \alpha_{2j} = \alpha_{goal}^{l}$ and $\alpha_{2i+1} = \alpha_{2j+1} = \alpha_{goal}^{h}$ $\forall i, j$ positive integers. So, we focus only on the dynamic behavior during the first fluctuation cycle $T_1 = 2(T_f + T_b) + T_q^{(1)} + T_d^{(1)} + T_l^{(1)} + T_r^{(1)}$ and the second fluctuation cycle $T_2 = 2(T_f + T_b) + T_q^{(2)} + T_d^{(2)} + T_l^{(2)} + T_r^{(2)}$, and define the *system period* to be $T = T_1 + T_2$.

In the i-th fluctuation cycle ($i = 1, 2$), let $R^{(i)}_{max}$ and $R^{(i)}_{min}$ be its the maximum rate and minimum rate, respectively, and $Q^{(i)}_{max}$ be its maximum queue length, then we have

$$R^{(i)}_{max} = \mu + \alpha_i(T^{(i)}_q + T_b + T_f) \tag{5.1}$$

where $T^{(i)}_q = \sqrt{\frac{2Q_h}{\alpha_i}}$ is the time for the queue length to grow from 0 to Q_h, $\alpha_1 = \alpha^h_{goal} = \alpha^l_{goal}/q$ and $\alpha_2 = q\alpha_1 = \alpha^l_{goal}$. For convenience of presentation, we define

$$T^{(i)}_{max} \stackrel{\triangle}{=} T_b + T^{(i)}_q + T_f = T_b + \sqrt{\frac{2Q_h}{\alpha_i}} + T_f \tag{5.2}$$

which is the time for $R(t)$ to increase from μ to its maximum $R^{(i)}_{max}$ by exercising linear rate-increase control. Then, the maximum queue length is expressed as

$$Q^{(i)}_{max} = \int_0^{T^{(i)}_{max}} \alpha_i t \; dt + \int_0^{T^{(i)}_d} (R^{(i)}_{max} e^{-(1-\beta_i)\frac{t}{\Delta}} - \mu) dt \tag{5.3}$$

where $T^{(i)}_d$ is the time for $R(t)$ to drop from $R^{(i)}_{max}$ to μ, and is obtained by letting $R^{(i)}(T_d) = \mu$ as:

$$T^{(i)}_d = -\frac{\Delta}{(1-\beta_i)} \; log \; \frac{\mu}{R^{(i)}_{max}}. \tag{5.4}$$

Then, we have

$$Q^{(i)}_{max} = \frac{\alpha_i}{2}[T^{(i)}_{max}]^2 + \alpha_i \frac{\Delta}{(1-\beta_i)} T^{(i)}_{max} + \mu \frac{\Delta}{(1-\beta_i)} \; log \; \frac{\mu}{R^{(i)}_{max}} \tag{5.5}$$

Note that the queue becomes empty during the fluctuations in equilibrium state as the utilization is $<$ 100 %. $T^{(i)}_l$ is the duration for $Q(t)$ to decrease from $Q^{(i)}_{max}$ to Q_l:

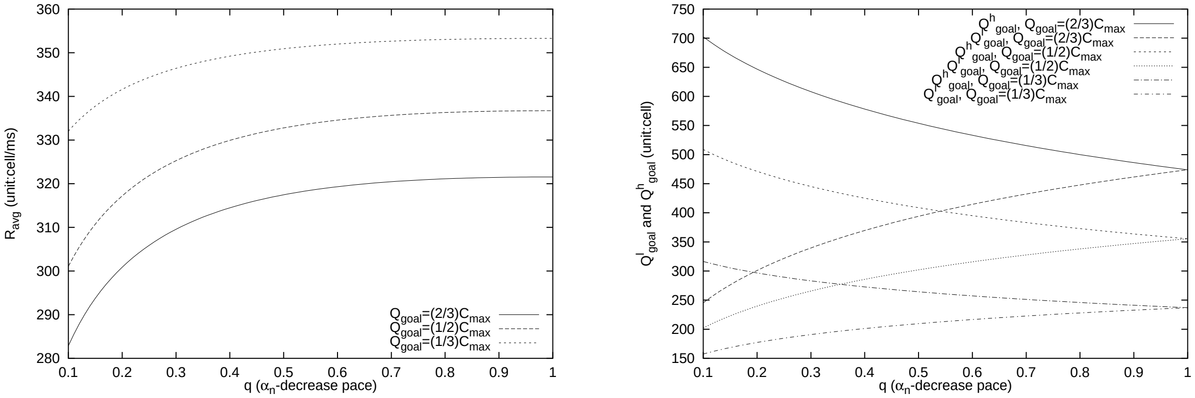
$$Q^{(i)}_{max} - Q_l = \int_0^{T^{(i)}_l} \mu(1 - e^{-(1-\beta_i)\frac{t}{\Delta}}) dt \tag{5.6}$$

So, $T^{(i)}_l$ is the non-negative real root of the nonlinear equation:

$$e^{-(1-\beta_i)\frac{T^{(i)}_l}{\Delta}} + \left(\frac{1-\beta_i}{\Delta}\right) T^{(i)}_l - \left[\left(\frac{Q^{(i)}_{max} - Q_l}{\mu}\right) \left(\frac{1-\beta_i}{\Delta}\right) + 1 \right] = 0. \tag{5.7}$$

The minimum rate is then given as $R^{(i)}_{min} = \mu e^{-(1-\beta_i)\frac{(T^{(i)}_l + T_b + T_f)}{\Delta}}$. The system period is

$$T = \sum_{i=1}^{2} T_i = \sum_{i=1}^{2} \Big[\; T^{(i)}_q + T^{(i)}_d + T^{(i)}_l + 2\tau + T^{(i)}_r \Big] \tag{5.8}$$



(a) Average throughput $\overline{R}$ vs. α_n-decrease pace q (b) Q^h_{goal} and Q^l_{goal} vs. α_n-decrease pace q

Figure 4: Average throughput and High/Low target buffer occupancy with $\mu = 367$ and $C_{max} = 711$.

where $T_r^{(i)} = (\mu - R_{min}^{(i)})/\alpha_{i+1}$ is the time for $R(t)$ to grow from $R_{min}^{(i)}$ to μ with the increase-rate parameter α_{i+1} ($\alpha_3 = \alpha_1$). Note that each T_i 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 equilibrium throughput can be calculated by averaging $R(t)$ over one cycle T as

$$\overline{R} = \frac{1}{T}\sum_{i=1}^{2}\Big[\int_0^{T_{max}^{(i)}}(\mu + \alpha_i t)dt + \int_0^{T_e^{(i)}}(R_{max}^{(i)}e^{-(1-\beta_i)\frac{t}{\Delta}})dt + \int_0^{T_r^{(i)}}(R_{min}^{(i)} + \alpha_{i+1}t)dt\Big] \tag{5.9}$$

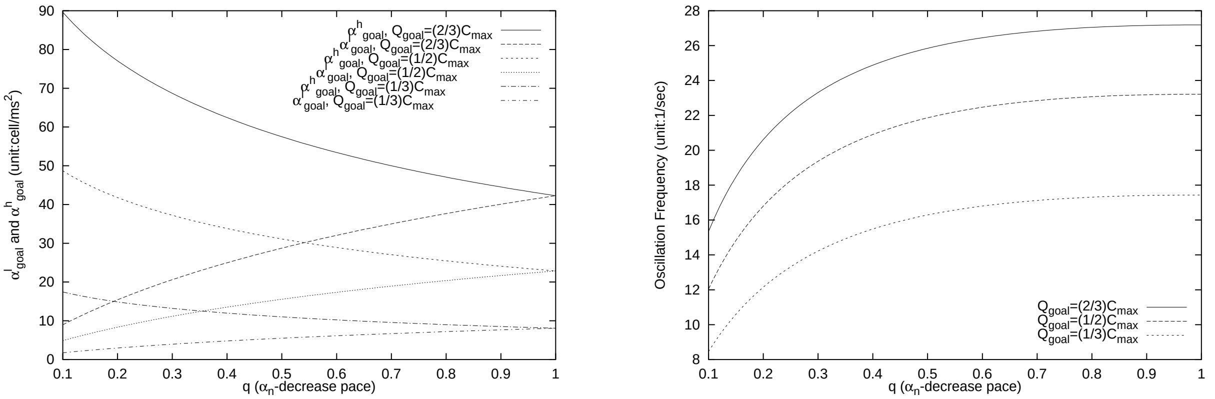
where $T_e^{(i)} = T_d^{(i)} + T_l^{(i)} + \tau$ is the time spent on exponential-decrease rate control within the i-th cycle. The above equation is reduced to:

$$\overline{R} = \frac{1}{T}\sum_{i=1}^{2}\Big[\mu T_{max}^{(i)} + \frac{\alpha_i}{2}[T_{max}^{(i)}]^2 + R_{max}^{(i)}\left(\frac{\Delta}{1-\beta_i}\right)\left(1 - e^{-(1-\beta_i)\frac{T_e^{(i)}}{\Delta}}\right) + T_r^{(i)}R_{min}^{(i)} + \frac{\alpha_{i+1}}{2}[T_r^{(i)}]^2\Big] \tag{5.10}$$

5.2 Numerical Evaluation of Equilibrium-State Performance

Using the analytical results derived thus far, we now compute equilibrium-state performance. We assumed that (i) the bottleneck link bandwidth $\mu = 155$ Mbps (367 cells/ms) and $C_{max} = 711$ cells, and (ii) the bottleneck is detected at a node farthest away from the source — the worst case in view of feedback delay — so $T_b = T_f = 1$ ms, thus $\tau = T_b + T_f = 2$ ms. Also, we use $\Delta = 0.5\tau = 1$ ms, $Q_h = 50$ cells, $Q_l = 25$ cells, and the initial source rate $R_0 = \mu$ as we are dealing with the equilibrium state.

First, we examine how the α-control parameter q affects $\overline{R}$. Figure 4(a) plots $\overline{R}$ vs. q for different values of Q_{goal}. According to the proposed α-control, we first focus on the ideal case where $Q_{goal} = \frac{1}{2}(Q^h_{goal} + Q^l_{goal})$, i.e., $Q_{max}^{(n)}$ fluctuates symmetrically above and below Q_{goal}. Figure 4(a) shows that $\overline{R}$ monotonically increases as q grows from 0.1 to 1.0. This is expected since a smaller q leads to a



(a) α^h_{goal} and α^l_{goal} vs. α_n-decrease pace q

(b) Oscillation frequency $\frac{1}{T}$ vs. α_n-decrease pace q.

Figure 5: High/Low target rate parameter and Oscillation frequency with $\mu = 367$ and $C_{max} = 711$.

larger fluctuation of $R^{(n)}_{max}$ and $Q^{(n)}_{max}$, which defeats the equilibrium-state performance of $\overline{R}$. When q gets larger, the fluctuation amplitudes of $Q^{(n)}_{max}$ and $R^{(n)}_{max}$ get smaller, as shown in Theorem 1. In the extreme case when $q \to 1$ (q cannot be equal to 1 since $q = 1$ means that the α-control is shut down), $R^{(n)}_{max}$ approaches a steady constant value, and the equilibrium-state performance of $\overline{R}$ attains its maximum. Figure 4(a) also indicates that for the same value of q, a smaller value of $Q_{goal} = kC_{max}$, $0 < k < 1$, leads to a larger $\overline{R}$ in equilibrium state since a smaller Q_{goal} implies a smaller α_{goal}. Figure 4(a) shows (i) a sharp drop in $\overline{R}$ when q gets smaller than 0.4, and (ii) a slow gain in $\overline{R}$ when $q > 0.6$, providing information on how to select q for the α-control to operate in a balanced region within which an optimal balance between average throughput and response speed is achieved.

Figure 4(b) plotted Q^h_{goal} and Q^l_{goal} against q for different values of Q_{goal}. Figure 4(b) shows that for a given Q_{goal}, Q^h_{goal} (Q^l_{goal}) is a monotonically decreasing (increasing) function of q. Moreover, as q increases, Q^l_{goal} and Q^h_{goal} approach each other symmetrically with respect to the given Q_{goal}; this is expected since $Q_{goal} = \frac{1}{2}(Q^h_{goal} + Q^l_{goal})$. As $q \to 1$, Q^h_{goal} will become close to Q^l_{goal}, thus resulting in almost no fluctuation in $Q^{(n)}_{max}$, which is consistent with the Eq. (4.4) in Theorem 1. Also, for a given q, a smaller setting of Q_{goal} (or α_{goal}) resulted in a smaller difference between Q^l_{goal} and Q^h_{goal}, thus a smaller fluctuation amplitude of $Q^{(n)}_{max}$, which also verifies Eq. (4.5) in Theorem 1.

Figure 5(a) plots α^l_{goal} and α^h_{goal} vs. q for different values of Q_{goal}, showing a similar pattern to that in Figure 4(b). However, as q increases, α^h_{goal} and α^l_{goal} approach each other asymmetrically, due to the nonlinear functionality of Q_{max} in α. Moreover, given q a larger Q_{goal} leads to a larger $(\alpha^h_{goal} - \alpha^l_{goal})$.

Selection of q also affects the oscillation frequency, $F \triangleq \frac{1}{T}$, of the flow-controlled system in the equilibrium state, where T is the oscillation cycle defined by Eq. (5.8). Figure 5(b) plots F against q

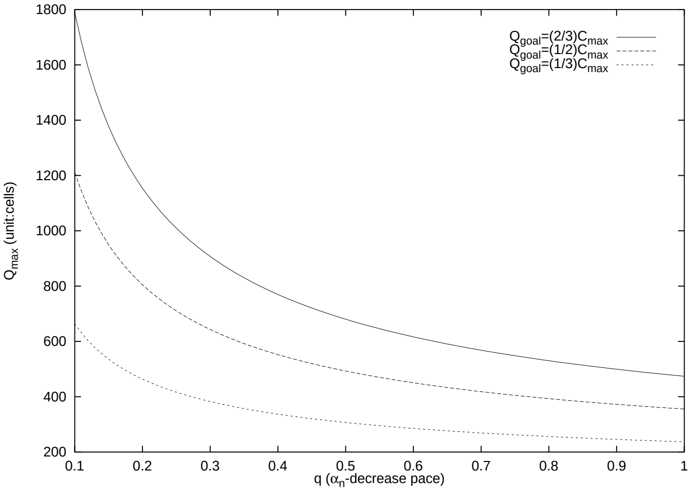


Figure 6: Maximum buffer occupancy Q_{max} vs. α_n-decrease pace q.

for different values of Q_{goal}. F is observed to monotonically increase with q regardless of Q_{goal} value. Thus, together with Figure 4(b), Figure 5(b) also shows that the oscillation frequency is inversely proportional to the oscillation amplitude of $Q^{(n)}_{max}$. Moreover, given q a larger Q_{goal} gives rise to a higher oscillation frequency.

In general, the ideal case of $Q_{goal} = \frac{1}{2}(Q^h_{goal} + Q^l_{goal})$ does not always hold, depending on the initial value of α_0, but the α-control algorithm guarantees the relationship $Q^l_{goal} \leq Q_{goal} \leq Q^h_{goal}$ to hold. In the more general case described by this relationship, $Q^{(n)}_{max}$ fluctuates around Q_{goal} after $Q^{(n)}_{max}$ converged to Q_{goal}'s closest neighborhood $\{Q^l_{goal}, Q^h_{goal}\}$. But Q_{goal} can be anywhere between Q^l_{goal} and Q^h_{goal}. To analyze how q affects the maximum buffer requirement, we consider the worst case when $Q_{goal} \overset{\sim}{\geq} Q^l_{goal}$. Figure 6 plots Q_{max} vs. q in the worst case of buffer requirement. Q_{max} is observed to increase as q decreases, which makes sense since a smaller q implies a larger fluctuation amplitude of $Q^{(n)}_{max}$. Moreover, when q is very small, particularly below the range of $0.4 \sim 0.6$, Q_{max} shoots up quickly. Also, when q is beyond the range of $0.4 \sim 0.6$, Q_{max} drops slowly as q increases.

5.3 Transient-State Analysis

An equilibrium state can be broken by either (1) the change of τ due to the change of system bottleneck location, or (2) the change of available bandwidth due to the variation in cross traffic. The transient state can be caused by the variation of τ in two different cases: (I) $\alpha_0 > \alpha^h_{goal}$, the rate convergence is under-damped, and (II) $\alpha_0 < \alpha^l_{goal}$, the rate convergence is over-damped, where α^h_{goal} and α^l_{goal} are specified by the target buffer occupancy Q_{goal}, and the α-control parameter q and p are selected and adjusted according to the variation of τ. Since the new system bottleneck usually has a smaller target bandwidth $\widetilde{\mu}$ than the old one μ, it is reasonable to assume that $R_0 > \widetilde{\mu}$ after the system bottleneck shifted to a different path. Thus, the previous system bottleneck path's target rate parameters become the new system bottleneck's initial values, denoted by α_0 and β_0. Let the new system bottleneck's target rate parameters be $\widetilde{\alpha}_{goal}$ which corresponds to the new system bottleneck's feedback delay $\widetilde{\tau}$. To quantitatively characterize transient-state performance, we define and determine an important performance parameter as follows.

Definition 4. *The number of transient cycles is defined by*

$$N \overset{\triangle}{=} \begin{cases} \max_{k\in\{0,1,2,\cdots\}}\{k \mid q^k\alpha_0 \geq \widetilde{\alpha}_{goal}\}; & if\ \alpha_0 \geq \widetilde{\alpha}_{goal} \\ \max_{k\in\{0,1,2,\cdots\}}\{k \mid \alpha_0 + kp \leq \widetilde{\alpha}_{goal}\}; & if\ \alpha_0 \leq \widetilde{\alpha}_{goal} \end{cases} \tag{5.11}$$

Theorem 2. *If the initial rate-control parameter $\alpha = \alpha_0$, the new system bottleneck feedback delay $\tau = \widetilde{\tau}$, and new system bottleneck target bandwidth $\mu = \widetilde{\mu}$, then N is determined by*

$$N = \begin{cases} \lfloor \frac{log\ [\frac{\widetilde{\alpha}_{goal}}{\alpha_0}]}{log\ q} \rfloor; & if\ \alpha_0 \geq \widetilde{\alpha}_{goal} \\ \lfloor \frac{\widetilde{\alpha}_{goal}-\alpha_0}{p} \rfloor; & if\ \alpha_0 \leq \widetilde{\alpha}_{goal} \end{cases} \tag{5.12}$$

where $\widetilde{\alpha}_{goal}$ is the non-negative real solution of

$$\frac{\widetilde{\alpha}_{goal}}{2}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha}_{goal}}}\right)^2 + \widetilde{\mu}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha}_{goal}}}\right) + \frac{\widetilde{\mu}^2}{\widetilde{\alpha}_{goal}}\ log\ \frac{\widetilde{\mu}}{\widetilde{\mu} + \widetilde{\alpha}_{goal}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha}_{goal}}}\right)} - Q_{goal} = 0 \tag{5.13}$$

and can be approximated as

$$\widetilde{\alpha}_{goal} \approx \left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\widetilde{\tau}}\right)^2. \tag{5.14}$$

Proof: The proof follows from the derivations of Eqs. (5.1) through Eq. (5.5) and definition of N given in Eq. (5.11). The proof for Eq. (5.14) is omitted here for lack of space.
□

Now let $R_{peak}^{(i)}$ and $Q_{peak}^{(i)}$ be the peak source rate and queue length, respectively, in the i-th transient cycle, $i = 1, 2, \cdots, N(\geq 1)$ (by assuming $\alpha_0 \geq \frac{1}{q}\widetilde{\alpha}_{goal}$ or $\alpha_0 \leq \widetilde{\alpha}_{goal} - p$). Let's start from the first transient cycle, or $i = 1$. Since the rate-increase function in the first transient cycle is $R(t) = R_0 + \alpha_0 t$, we have

$$R_{peak}^{(1)} = R_0 + \alpha_0(T_q^{(1)} + \widetilde{\tau}) \tag{5.15}$$

where $T_q^{(1)} = \frac{1}{\alpha_0}\left[-(R_0 - \widetilde{\mu}) + \sqrt{(R_0 - \widetilde{\mu})^2 + 2\alpha_0 Q_h}\right]$ is obtained by solving $Q_h = \int_0^{T_q^{(1)}}(R(t) - \widetilde{\mu})dt$. For convenience, let $T_{peak}^{(1)} \overset{\triangle}{=} T_q^{(1)} + \widetilde{\tau}$ be the time for $R(t)$ to increase from R_0 to $R_{peak}^{(1)}$. Then,

$$Q_{peak}^{(1)} = \int_0^{T_{peak}^{(1)}}(R_0 + \alpha_0 t - \widetilde{\mu})dt + \int_0^{T_d^{(1)}}(R_{peak}^{(1)} e^{-(1-\beta_0)\frac{t}{\Delta}} - \widetilde{\mu})dt \tag{5.16}$$

where $T_d^{(1)} = -\frac{\Delta}{(1-\beta_0)} \; log \; \frac{\tilde{\mu}}{R^{(1)}_{peak}}$ is the time for $R(t)$ to drop from R_{peak} back to $\tilde{\mu}$. Reducing Eq. (5.16) gives

$$Q^{(1)}_{peak} = (R_0 - \tilde{\mu})T^{(1)}_{peak} + \frac{\alpha_0}{2}[T^{(1)}_{peak}]^2 + \tilde{\mu}\frac{\Delta}{1-\beta_0}\left[\frac{\alpha_0 T^{(1)}_{peak}}{\tilde{\mu}} + \left(\frac{R_0}{\tilde{\mu}} - 1\right) + log\;\frac{\tilde{\mu}}{R^{(1)}_{peak}}\right] \tag{5.17}$$

When $R_0 = \tilde{\mu}$, Eq. (5.17) reduces to Eq. (5.5), which is consistent with the fact that $Q^{(i)}_{max}$ is the special case of $Q^{(1)}_{peak}$ with $R_0 = \tilde{\mu}$.

To compute the first transient-state cycle, we need to find $T_l^{(1)}$ which is the non-negative real root of nonlinear equation:

$$e^{-(1-\beta_0)\frac{T_l^{(1)}}{\Delta}} + \left(\frac{1-\beta_0}{\Delta}\right)T_l^{(1)} - \left[\left(\frac{Q^{(1)}_{peak} - Q_l}{\tilde{\mu}}\right)\left(\frac{1-\beta_0}{\Delta}\right) + 1\right] = 0. \tag{5.18}$$

This transient-state cycle is $T^{(1)} = T_q^{(1)} + T_d^{(1)} + T_l^{(1)} + 2\tilde{\tau} + T_r^{(1)}$ where $T_r^{(1)} = \frac{\tilde{\mu}}{\alpha_1}\left(1 - e^{-(1-\beta_0)\frac{T_l^{(1)}+\tilde{\tau}}{\Delta}}\right)$ is the time for $R(t)$ to reach $\tilde{\mu}$ from its lowest value in the first transient cycle.

Finally, the average throughput during the first transient-state cycle is expressed by

$$\begin{aligned}\overline{R}^{(1)} &= \frac{1}{T^{(1)}}\Bigg[\; R_0 T^{(1)}_{peak} + \frac{\alpha_0}{2}[T^{(1)}_{peak}]^2 + R^{(1)}_{peak}\left(\frac{\Delta}{1-\beta_0}\right)\left(1 - e^{-(1-\beta_0)\frac{T_d^{(1)}+T_l^{(1)}+\tilde{\tau}}{\Delta}}\right) \\ &\quad + T_r^{(1)}\left(\tilde{\mu}e^{-(1-\beta_0)\frac{T_l^{(1)}+\tilde{\tau}}{\Delta}}\right) + \frac{\alpha_1}{2}[T_r^{(1)}]^2\Bigg]\end{aligned} \tag{5.19}$$

Now, let's consider cases for $2 \leq i \leq N$; since the performance parameters are derived similarly to the case of $i = 1$, we only give the final expressions for the average throughput, the peak queue length, and the length of the i-th transient cycle ($2 \leq i \leq N$):

$$\begin{aligned}\overline{R}^{(i)} &= \frac{1}{T^{(i)}}\Bigg[\; \tilde{\mu}T^{(i)}_{peak} + \frac{\alpha_{i-1}}{2}[T^{(i)}_{peak}]^2 + R^{(i)}_{peak}\left(\frac{\Delta}{1-\beta_{i-1}}\right)\left(1 - e^{-(1-\beta_{i-1})\frac{T_d^{(i)}+T_l^{(i)}+\tilde{\tau}}{\Delta}}\right) \\ &\quad + T_r^{(i)}\left(\tilde{\mu}e^{-(1-\beta_{i-1})\frac{T_l^{(i)}+\tilde{\tau}}{\Delta}}\right) + \frac{\alpha_i}{2}[T_r^{(i)}]^2\Bigg]\end{aligned} \tag{5.20}$$

$$Q^{(i)}_{peak} = \frac{\alpha_{i-1}}{2}[T^{(i)}_{peak}]^2 + \alpha_{i-1}\frac{\Delta}{(1-\beta_{i-1})}T^{(i)}_{peak} + \tilde{\mu}\frac{\Delta}{(1-\beta_{i-1})}\; log\; \frac{\tilde{\mu}}{R^{(i)}_{peak}} \tag{5.21}$$

$$T^{(i)} = \sqrt{\frac{2Q_h}{\alpha_{i-1}}} + T_d^{(i)} + T_l^{(i)} + T_r^{(i)} + 2\tilde{\tau} \tag{5.22}$$

where

$$
\begin{aligned}
T_{peak}^{(i)} &= \tilde{\tau} + \sqrt{\frac{2Q_h}{\alpha_{i-1}}} \\
R_{peak}^{(i)} &= \tilde{\mu} + \alpha_{i-1} T_{peak}^{(i)} \\
T_d^{(i)} &= -\frac{\Delta}{(1-\beta_{i-1})}\, log\, \frac{\tilde{\mu}}{R_{peak}^{(i)}} \\
T_r^{(i)} &= \frac{\tilde{\mu}}{\alpha_i}\left(1 - e^{-(1-\beta_{i-1})\frac{T_l^{(i)}+\tilde{\tau}}{\Delta}}\right)
\end{aligned}
$$

and $T_l^{(i)}$ is the non-negative real root of the following nonlinear equation

$$
e^{-(1-\beta_{i-1})\frac{T_l^{(i)}}{\Delta}} + \left(\frac{1-\beta_{i-1}}{\Delta}\right) T_l^{(i)} - \left[\left(\frac{Q_{peak}^{(i)} - Q_l}{\tilde{\mu}}\right)\left(\frac{1-\beta_{i-1}}{\Delta}\right) + 1\right] = 0. \qquad (5.23)
$$

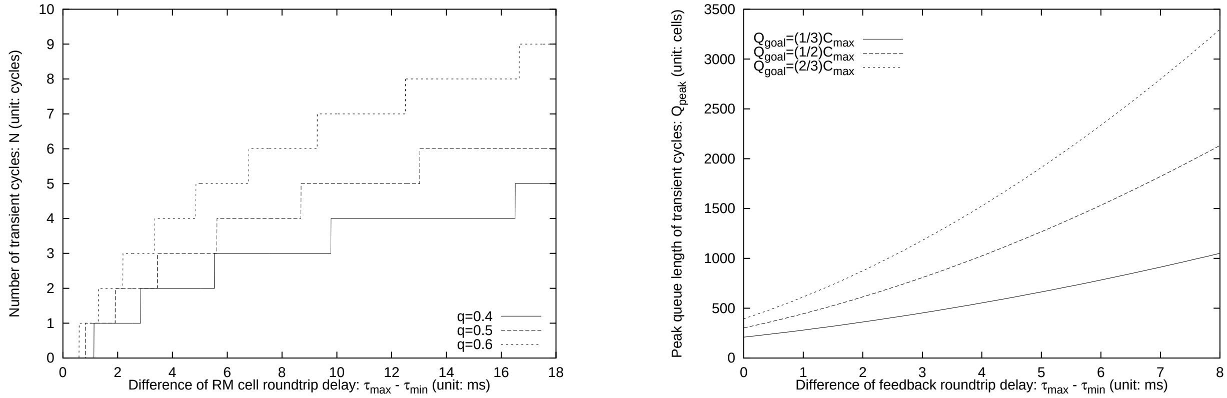
The entire transient-state period is then $T_{tran} = \sum_{i=1}^{N} T^{(i)}$, and its average throughput is expressed by

$$
\overline{R}_{tran} = \frac{1}{T_{tran}} \sum_{i=1}^{N} \overline{R}^{(i)} T^{(i)}. \qquad (5.24)
$$

The peak queue length for the case of $\alpha_0 > \alpha_{goal}^{h}$ is $Q_{peak} = Q_{peak}^{(1)}$. Here N is defined by Eq. (5.11) and α_i by the α-control law Eq. (4.1).

5.4 Numerical Evaluation of Transient-State Performance

Based on the analysis in Section 5.3, we derived numerical results of transient-state performance. We use the same network flow-control settings as in the equilibrium-state analysis: $Q_h = 50$ cells, $Q_l = 25$ cells, $\Delta = 1$ ms. But for the transient-state flow control, we set $C_{max} = 700$ cells and $Q_{goal} = \frac{1}{2} C_{max} = 350$ cells, and the previous equilibrium state is specified by $\mu_0 = 367$ cells/ms and initial system bottleneck feedback delay $\tau_0 = 2$ ms. We use the new system bottleneck target available bandwidth $\tilde{\mu} = 267$, and focus on the worst case when the system bottleneck moves from the shortest path to the longest one. So, the system bottleneck feedback delay changes from $\tau_0 = \tau_{min} \triangleq \min_{i \in \{1,2,\cdots,n\}}\{\tau_i\}$ to $\tilde{\tau} = \tau_{max} \triangleq \max_{i \in \{1,2,\cdots,n\}}\{\tau_i\}$, for an n-branch pt-to-mpt connection. Figure 7(a) plots N, the number of cycles in the transient state, versus $(\tau_{max} - \tau_{min})$ for different values of q. N is found to increase stepwise monotonically with $(\tau_{max} - \tau_{min})$. This was expected since a large change in system bottleneck feedback delay requires more transient cycles to converge to the new optimal equilibrium state. A smaller q results in a fewer number of transient cycles. Thus, q measures the speed of convergence. When $\tau_{max} = 10\tau_{min}$ and $q = 0.4$, the flow control takes only 5 cycles to converge to the new optimal equilibrium state.



(a) Number of transient cycles: N vs. $(\tau_{max} - \tau_{min})$ (b) Peak queue length: Q_{peak} vs. $(\tau_{max} - \tau_{min})$

Figure 7: Number of transient cycles and Transient-state peak queue length.

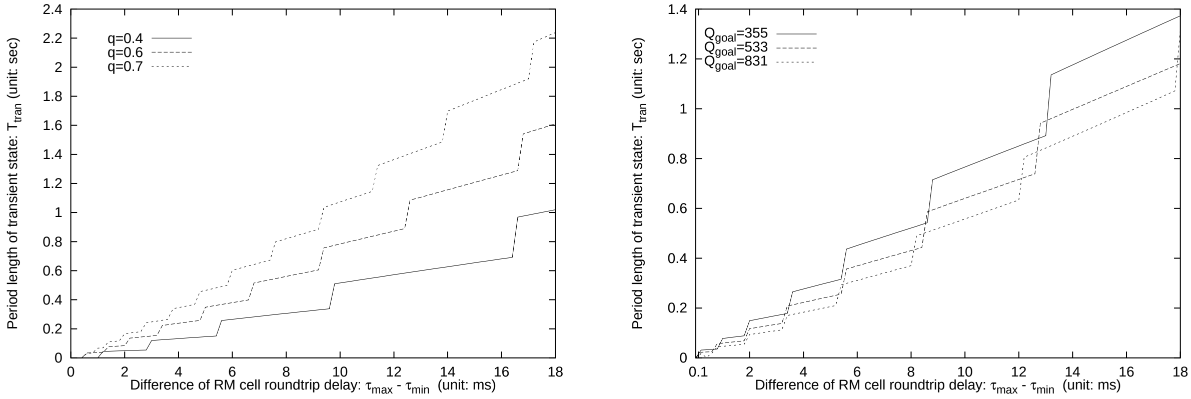
Figure 7(b) plots Q_{peak} vs. $(\tau_{max} - \tau_{min})$ for a different target buffer occupancy Q_{goal}, and $R_0 = 367$, $\widetilde{\mu} = 267$, $\tau_0 = 2$ ms, $Q_h = 50$ cells, $Q_l = 25$ cells, and $\Delta = 1$ ms. Q_{peak} is observed to shoot up quickly with $(\tau_{max} - \tau_{min})$ and a larger target buffer occupancy is found to result in a faster increase of Q_{peak}. Figure 8(a) plots the period of transient state T_{tran} versus $(\tau_{max} - \tau_{min})$ for different values of q. T_{tran} increases piecewise linearly with $(\tau_{max} - \tau_{min})$. Given $(\tau_{max} - \tau_{min})$, a smaller q results in a faster transient convergence (shorter T_{tran}), which is also expected since q is a measure of α-decrease rate. A small q implies a large α-decrease pace. Furthermore, while T_{tran} decreases as q gets smaller, $T^{(i)}$ increases in each transient cycle, leading to a smaller N. In general, a smaller q results in faster convergence. Figure 8(b) shows that for the given $(\tau_{max} - \tau_{min})$ and N, the smaller Q_{goal} is, the slower the convergence process is.

6 Multiple Multicast Connections

We now analyze the performance of the proposed scheme for N (> 1) concurrent multicast connections that share a common system bottleneck.

6.1 System Model

N concurrent flow-controlled connections with a common system bottleneck are modeled by a single buffer and a server shared by N sources as shown in Figure 9. The parameters characterizing the i-th multicast connection $(i = 1, 2, \cdots, N)$ are given as follows.



(a) T_{tran} vs. $(\tau_{max} - \tau_{min})$ with different qs (b) T_{tran} vs. $(\tau_{max} - \tau_{min})$ with different Q_{goal}s

Figure 8: Transient-state period for different qs and Q_{goal}s

$R_i(t)$: Data transmission rate for the i-th multicast connection
$\alpha^{\langle i \rangle}$: Additive rate-increase parameter for the i-th multicast connection
$\beta^{\langle i \rangle}$: Multiplicative decrease factor for the i-th multicast connection
Δ_i: Time interval of rate update for the i-th multicast connection
$T_f^{\langle i \rangle}$ ($T_b^{\langle i \rangle}$): Forward (backward) delay for the i-th multicast connection
$Q(t)$: Length of the shared queue at the system bottleneck
Q_h (Q_l): High (low) threshold of the system-bottleneck queue
μ: System-bottleneck link bandwidth (BW)

We now derive the equations describing the dynamical behavior of this system by proceeding as in Section 3. At time t the aggregate arrival rate at the system bottleneck is $\sum_{i=1}^{N} R_i(t - T_f^{\langle i \rangle})$, so

$$Q(t) = \begin{cases} 0; & \text{if } Q(t) = 0 \wedge \sum_{i=1}^{N} R_i(t) < \mu \\ \int_{t_0}^{t} \left\{ \sum_{i=1}^{N} R_i(v - T_f^{\langle i \rangle}) - \mu \right\} dv + Q(t_0); & \text{if (1) } \sum_{i=1}^{N} R_i(t) > \mu\text{; or} \\ & \quad \text{(2) } \sum_{i=1}^{N} R_i(t) < \mu \wedge Q(t) > 0 \end{cases} \tag{6.1}$$

Applying the same rate-control algorithm proposed in Section 2, for $i = 1, 2, \cdots, N$, we get:

$$R_i(t) \quad = \quad \begin{cases} R_i(t_0) + \alpha^{\langle i \rangle}(t - t_0); & \text{if } Q(t - T_b^{\langle i \rangle}) < Q_h \\ R_i(t_0) e^{-(1-\beta^{\langle i \rangle})\frac{(t-t_0)}{\Delta_i}}; & \text{if } Q(t - T_b^{\langle i \rangle}) \geq Q_h \end{cases} \tag{6.2}$$

The α-control is applied in the same fashion as in the single multicast connection case, but $Q_{max}^{(n)}$ is contributed, and Q_{goal} is shared, by all N connections.

Derivation of analytical results for multiple concurrent multicast connections is quite lengthy and thus omitted. Using these analytical results, we now present two examples to demonstrate the efficacy of the proposed scheme in terms of convergence to target buffer occupancy and link bandwidth, and fairness of buffer and link bandwidth sharing.

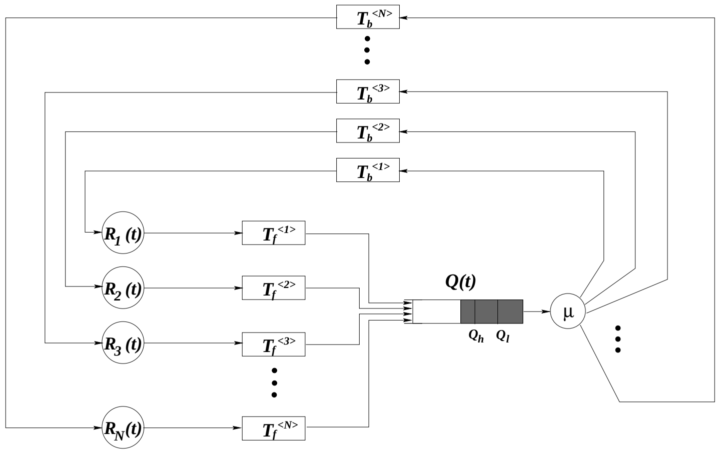


Figure 9: The system model for N concurrent connections sharing a system bottleneck.

6.2 Adaptations to Variation in Feedback Delay and Available Bandwidth

In the first example, we consider 3 multicast connections VC$_1$, VC$_2$, VC$_3$, such that VC$_1$ and VC$_2$ first share the system bottleneck $\mathcal{B}_1$, then VC$_1$ and VC$_3$ share the system bottleneck $\mathcal{B}_2$ after the system bottleneck shifts from $\mathcal{B}_1$ to $\mathcal{B}_2$. $\mathcal{B}_1$'s parameters are $\mu = 45$ cells/ms, $\tau = 3$ ms (assumed the same for both VCs for simplicity), $\Delta = 0.4$ ms, $C_{max} = 500$ cells, $Q_{goal} = 250$ cells, $Q_h = 50$ cells, $Q_l = 25$ cells. We assume that $R_1(t)$ and $R_2(t)$ have already entered their equilibrium states with parameters $q_1 = q_2 = 0.5$, $\alpha_{goal}^{l\langle 1\rangle} = 2$ cells/ms^2, $\alpha_{goal}^{h\langle 1\rangle} = 1$ cell/ms^2, $\alpha_{goal}^{l\langle 2\rangle} = 1$ cell/ms^2, $\alpha_{goal}^{h\langle 2\rangle} = 0.5$ cell/ms^2, $MCR_1 = MCR_2 = 0$. At time $t = 325.66$ ms, the system bottleneck shifts from $\mathcal{B}_1$ to $\mathcal{B}_2$. The system bottleneck feedback delay of $\mathcal{B}_2$ is doubled to $\tau = 6$ ms while the available bandwidth decreases to $\mu = 40$ cells/ms (smaller than $\mathcal{B}_1$'s), and all the other parameters remain the same as $\mathcal{B}_1$'s. In $\mathcal{B}_2$, we assume that VC$_1$ shares the buffer and bandwidth with VC$_3$, and $R_1(t)$ starts competing for bandwidth with a higher steady value $\alpha^{\langle 2\rangle} = 2$ cells/ms^2 at the end of α-control cycle with a lower steady value of α and $R_3(t)$ with initial $\alpha^{\langle 3\rangle} = 1$ cell/ms^2 and $R_0^{\langle 3\rangle} = 20$ cells/ms. Using the analytical results for multiple concurrent multicast connections, we obtained the evolution functions of $R_1(t)$ and $R_3(t)$ and their shared $Q(t)$ with and without α-control, as shown in Figures 10 and 11, respectively,

For the scheme with α-control, we observe from Figure 10 that while in equilibrium state, $Q_{max}^{(n)}$ fluctuates periodically around its target buffer occupancy $Q_{goal} = 250$. At system bottleneck transition time $t = 325.66$ ms, Figure 10 shows that the shared $Q_{max}^{(n)}$ shoots up to $Q_{peak} = 472$ due to the doubling of system bottleneck feedback delay τ and decreasing of available bandwidth. By applying the α-control, $Q_{max}^{(n)}$ converges to, and stays in, its target's neighborhood within 3 cycles. Moreover, the proposed scheme ensures that $R_1(t)$ and $R_3(t)$ converge to their target bandwidth share, $\mu_{goal}^{\langle 1\rangle} = \mu_{goal}^{\langle 3\rangle} = 20$ cells/ms in a fair manner. The resulting bottleneck average bandwidth utilization in the equilibrium state is $\overline{R_1}/\mu_{goal}^{\langle 1\rangle} = 0.986$ and $\overline{R_3}/\mu_{goal}^{\langle 3\rangle} = 0.967$, respectively.

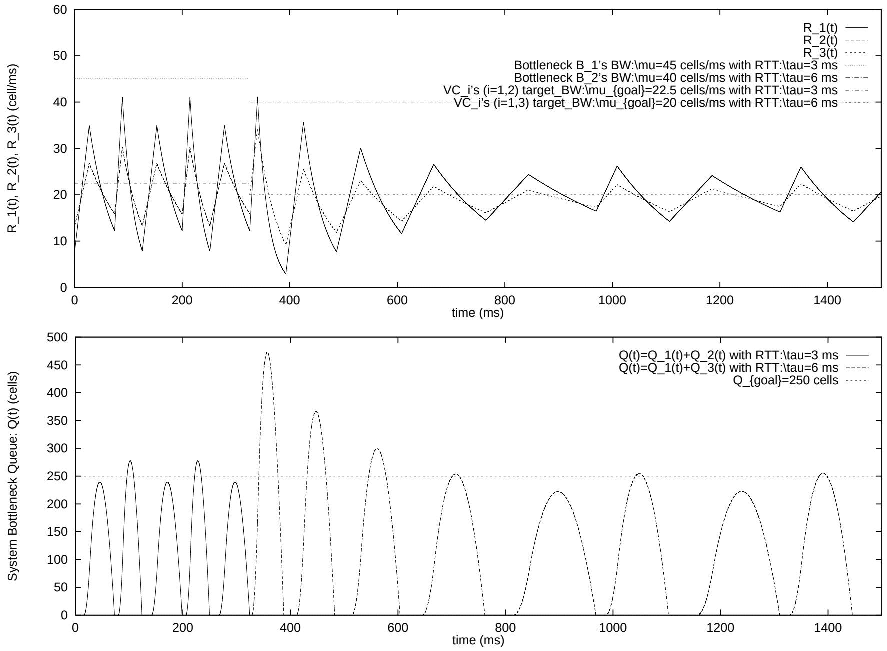


Figure 10: $Q_{max}^{(n)}$ adapts to system bottleneck feedback delay variation with α-control.

By contrast, for the scheme without α-control, Figure 11 illustrates that it cannot control $Q_{max}^{(n)}$ shoot-up when the system bottleneck feedback delay increases and available bandwidth decreases. We assume the same network and flow-control conditions for the scheme without α-control and $Q_{max}^{(n)}$ has been within Q_{goal}'s neighborhood since the system entered equilibrium state initially. When the system bottleneck shifts from $\mathcal{B}_1$ to $\mathcal{B}_2$ at time $t = 325.66$ ms, Figure 11 shows that $Q_{max}^{(n)}$ jumps up to $Q_{peak} = 476$ cells while in the transient state, and stays at as high as $Q_{max}^{(n)} = 461$ cells in the equilibrium state, which is almost two times of its setpoint $Q_{goal} = 250$ cells. However, total average bandwidth utilization in the equilibrium state is $\overline{R_1}/\mu_{goal}^{\langle 1 \rangle} = 0.951$ and $\overline{R_3}/\mu_{goal}^{\langle 3 \rangle} = 0.863$, respectively, which is lower than the α-controlled scheme.

This example indicates that our proposed scheme adapts well to the variations of system bottleneck feedback delay and available bandwidth of system bottleneck of a multicast connection while the scheme without α-control does not. Thus, our scheme scales well in terms of buffer requirement and average throughput as compared to the existing schemes.

6.3 Fairness of System Bottleneck Buffer Occupancy and Link Bandwidth

In the second example, we consider two multicast connections VC$_1$ and VC$_2$ sharing a single system bottleneck $\mathcal{B}_1$. We assume $R_1(t)$ of VC$_1$ start ramping up from $R_0^{\langle 1 \rangle} = 0$ with $\alpha_0^{\langle 1 \rangle} = 2$ cells/ms^2, $p^{\langle 1 \rangle}$=2 cells/ms^2, and $q^{\langle 1 \rangle}$=0.6. $\mathcal{B}_1$'s parameters are kept the same as those of $\mathcal{B}_1$'s in the first example. VC$_2$ joins VC$_1$ after VC$_1$ has reached its equilibrium state.

We assume VC$_1$ starts sending data at $t = 0$, and VC$_2$ at $t = 250.23$ ms which equals 3 α-

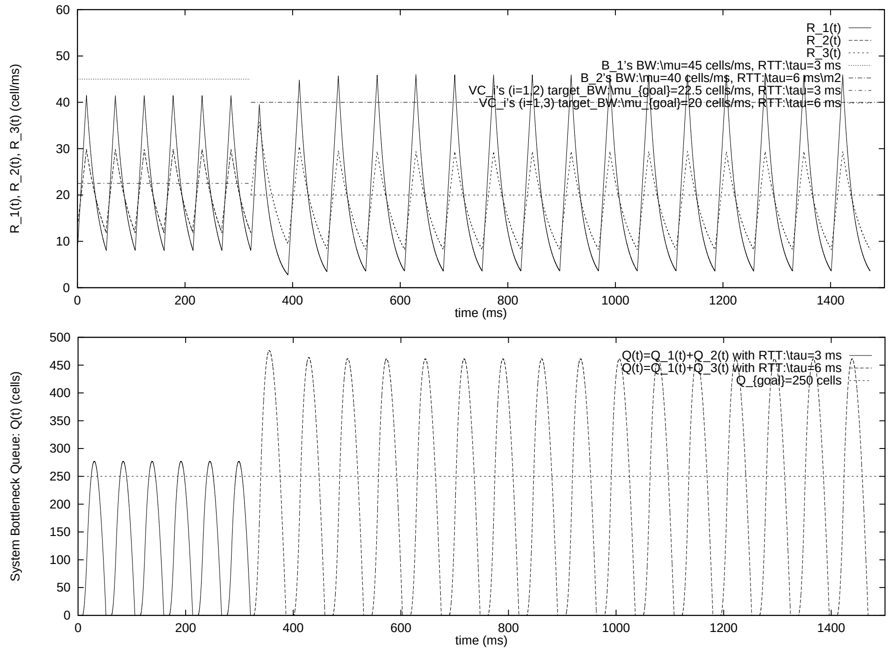


Figure 11: $Q_{max}^{(n)}$ does not adapt to system bottleneck feedback delay variation without α-control.

control cycles after VC$_1$ has already reached its equilibrium state. For the proposed scheme, using the analytical results derived from multiple multicast connection model, we computed the evolutions of $Q_1(t)$, $Q_2(t)$, and $Q(t) = Q_1(t)+Q_2(t)$ for both transient and equilibrium states as shown in Figure 12. After 4 transient cycles from $t = 0$, $Q_{max}^{(n)}$ is observed to converge to the neighborhood of $Q_{goal} = 250$ cells, instead of its share of $Q_{goal}^{\langle 1 \rangle} = \frac{1}{2}Q_{goal} = 125$ cells, because there are no other VCs sharing C_{max} with VC$_1$, and thus, VC$_1$ grabs the entire $Q_{goal} = 250$. At $t = 250.23$ ms, VC$_2$ starts competing for use of $\mathcal{B}_1$'s bandwidth capacity μ and buffer capacity C_{max}. At the same time, the equilibrium state of VC$_1$ is broken and $Q_1(t)$ and $R_1(t)$ start to give up the link bandwidth and buffer occupancy above their shares: $\mu_1 = 22.5$ cells/ms and $Q_{goal}^{\langle 1 \rangle} = 125$ cells. Note that right after VC$_2$ joins in $\mathcal{B}_1$, there is a transient period during which $R(t) = R_1(t) + R_2(t)$ and $Q(t) = Q_1(t) + Q_2(t)$ overtake their target values. After 4 transient cycles, the proposed flow-control scheme not only brings $R_1(t)$, $R_2(t)$ towards to their shares $\mu_i = \frac{1}{2}\mu = 22.5$ cells/ms^2 $(i = 1, 2)$, but also makes $Q_{max}^{(n)}$'s of $Q_1(t)$ and $Q_2(t)$ converge to their target shares: $Q_{goal}^{\langle i \rangle} = \frac{1}{2}Q_{goal} = 125$ cells $(i = 1, 2)$ as shown in Figure 12. Especially, the ratio of $Q_{max}^{(n)}$ shares between VC$_1$'s and VC$_2$'s is improved on average from 64% : 36% in transient state to 53% : 47% in equilibrium state.

This example shows that the proposed scheme is fair among the competing multicast connections in term of bandwidth and buffer capacity. In the scheme without α-control, since α which determines both target bandwidth and buffer occupancy is fixed, the fairness for both bandwidth and buffer occupancy is difficult to achieve for multiple multicast connections with different α's.

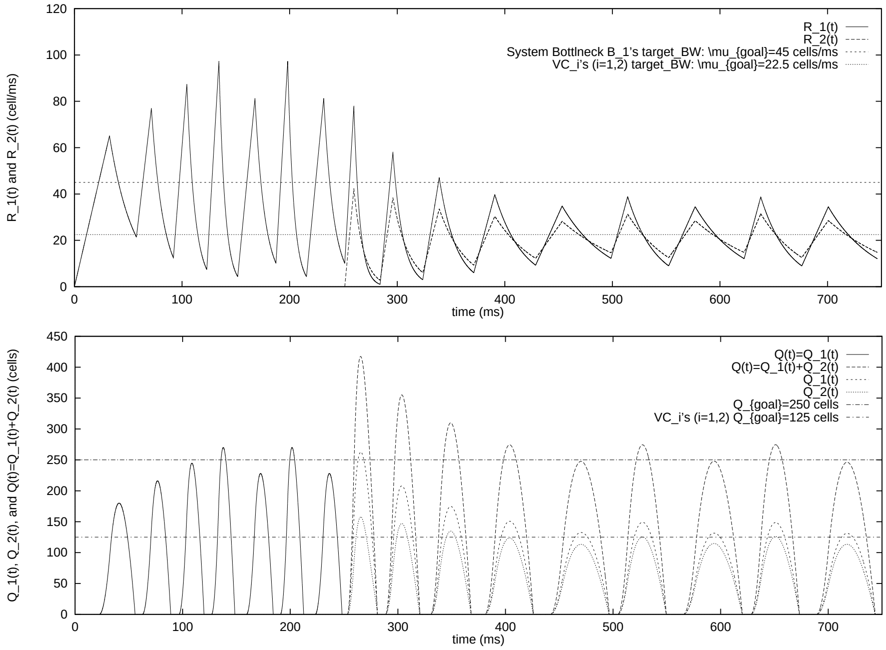


Figure 12: Fairness of bandwidth and buffer occupancy with α-control.

7 Conclusion

In this paper we proposed and evaluated a flow-control scheme for ATM ABR multicast services, which scales well and is efficient in dealing with the variations in the multicast-tree structure and feedback round trip delay. We proposed a second-order rate control algorithm to handle the variation of feedback round trip delay. By exercising two-dimensional rate control, the proposed scheme not only makes the transmission rate converge to the available bandwidth of the connection's most congested branch, but also brings the buffer occupancy to a small neighborhood of the target setpoint bounded by buffer capacity.

Using the fluid approximation, we modeled the proposed flow-control scheme and analyzed the system dynamics under the most stressful traffic condition. We derived closed-form expressions for queue buildups, average throughput, and other measures. These expressions were then used to evaluate the system performance, design the optimal rate-control parameters, and compute the evolution of the rate and queue-length functions. We also analyzed the convergence property of the second-order rate control law.

Using numerical examples, we have shown that the proposed scheme can achieve fairness among competing multicast connections in terms of both bandwidth and buffer capacity. We are currently developing a simulator based on NetSim to further evaluate the performance for more complicated network configurations under the proposed flow control, which will be reported in a forthcoming paper.

References

[1] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *Proc. of ACM SIGCOMM*, pp. 247–256, August 1988.

[2] L. Roberts, *Rate Based Algorithm for Point to Multipoint ABR Service*, ATM Forum contribution 94-0772, September 1994.

[3] L. Roberts, *Point-to-Multipoint ABR Operation*, ATM Forum contribution 95-0834, August 1995.

[4] K.-Y. Siu and H.-Y. Tzeng, "Congestion control for multicast service in ATM networks," in *Proc. of GLOBECOM*, pp. 310–314, November 1995.

[5] K.-Y. Siu and H.-Y. Tzeng, "On max-min fair congestion control for multicast ABR services in ATM," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 545–556, April 1997.

[6] H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe, "Performance issues in public ABR service," *IEEE Communications magazine*, vol. 11, pp. 40–48, November 1996.

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[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] 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.

[17] X. Zhang, K. G. Shin, and Q. Zheng, "Integrated rate and credit feedback control for ABR services in ATM networks," in *Proc. of IEEE INFOCOM*, April 1997.