Performance Analysis of Multicast Flow Control Algorithms over Combined Wired/Wireless Networks

Huayan Wang, *Student Member, IEEE,* and Mischa Schwartz, *Life Fellow, IEEE*

***Abstract*— A multicast flow control framework for data traffic traversing both a wired and wireless network is proposed. Markov-modulated fluid (MMF) models are used for the receivers to capture the dynamics of the wireless links. Our study shows that the phase differences of the instantaneous throughput capabilities of the receivers are a distinctive feature of multicast connections. The objectives of the multicast flow control algorithms are to cope with the receiver phase differences (RPD's) cost effectively in addition to the general goals such as maximizing throughput and minimizing delay. Three ad hoc algorithms have been studied: listen to slowest request (LSQ), source estimation (SE), and open-loop control. A fluid-flow analysis technique is applied to study the effect of receiver phase differences assuming zero propagation delay. The effect of propagation delay in multicast connections is then discussed. Simulation results are presented to verify the analysis for the zero-delay case and to compare the performance of the algorithms under nonnegligible delays. It turns out that the zero-delay case reveals the characteristics of the multicast algorithms and provides good performance bounds for the cases with nonnegligible propagation delays.**

***Index Terms*— Fluid flow analysis, integrated wired/wireless networks, multicast flow control.**

I. Introduction

FLOW control has been an important and active area of research for many years. Most of the schemes developed or investigated have been for point-to-point communications (unicast). They have generally focused on one type of traffic, and have assumed a wired, relatively high-speed communication infrastructure. Interest in the past few years has begun to shift to universal personal communication, involving both wired and wireless transmission media, to users on the move, to multimedia communication, and, as one prominent set of applications, to the transmission of the information to multiple users/receivers (multicast) [1].

In the past few years, numerous research projects have been carried out to explore how to provide multicast service efficiently and effectively in various network infrastructures. These include prototype systems supplying multicast communications with data and multimedia traffic [2]–[4], reliable and unreliable multicast protocols over LAN [5]–[7], Internet [8]–[13], ATM [14], [15], and extensions to include mobile hosts [16]. Much of this work is experimental in nature, and not many authors have done in-depth studies of the multicast flow control problem. The proposed/implemented multicast flow control mechanisms in the above systems/protocols can be summarized into three main categories. 1) Many of them extend the TCP window flow control protocol to multicast connections in various ways [2]–[4], [7]–[9], [13]. The basic idea is to coalesce the feedback information from all receivers into a single response. For example, a packet is not considered fully acknowledged until all receivers have sent an acknowledgment. There are different proposals on how to estimate the retransmission timeout period, and how to increase or decrease the window size. 2) Some of them adopt rate-based flow control. The source increases or decreases the transmission rate based on the feedback requests from the receivers [5], [10], [14], [15]. 3) Some of them use the hybrid rate- and window-based flow control approach. That is, they use the window flow control with a maximum rate limit [11], [12] to avoid congestion.

Recently, much effort has been devoted to the topic of consolidating the feedback signals (ACK's or NACK's) to avoid the feedback implosion problem [11], [15]. But few authors have addressed various ways of controlling the source transmission rate. Much of the above-referenced work uses some kind of "*listen to the slowest request*" strategy, by which we mean that the multicast flow control is based on some unicast protocol (such as TCP flow control), and the source acts in response to the slowest request. For example, the control window is not advanced if at least one receiver has not responded with a positive ACK. Similarly, in the rate-based control case, the source rate decreases if at least one receiver requests a rate reduction. These strategies lead to slowing down all receivers to the speed of the slowest receiver ["*listen to the slowest receiver*" (LSR)] if the receivers have constant throughput capability. In a dynamic environment where the receiver throughput capability varies, this type of strategy results in a throughput of a multicast connection that will be roughly the lowest profile among all receivers' instantaneous throughput capability curves. The available bandwidth at the receivers in a good throughput state is wasted if there is at least one receiver in the same multicast session in a bad throughput state. This is undesirable, especially for mobile receivers because wireless bandwidth is expensive. Nevertheless, the "listen to the slowest request" approach is widely implemented in existing systems because of its *simplicity* and *guaranteed stability* provided the corresponding point-to-point flow control protocol is stable. But more efficient algorithms are desired [11], [17].

Manuscript received August 30, 1996; revised March 20, 1997. This work was supported by the U.S. Army Research Office under Grant DAAH04-95-1-0188. This paper was presented in part at IEEE INFOCOM'97, Kobe, Japan, April 1997.

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

Publisher Item Identifier S 0733-8716(97)05843-5.

In addition to the "listen to the slowest request" approach, some other ways of combining the feedback requests from multiple receivers could be "weighted sum" [18], "voting" [19], or "random listening." On the other hand, we think that it is worth exploring some *multicast-aware* flow control algorithms, that is, to design new algorithms optimized for multicasting. To do so, we need to understand the features of multicast connections which affect flow control. We have not seen any work reported to explore these possible schemes and to compare the tradeoffs of the different policies. Reference [18] mentions "weighted sum" as one possible approach without further evaluation. "Voting" is proposed in [19] for the best effort delivery of video traffic in the Internet; the impact of voting threshold on the performance and the optimal design of the algorithm were left as open problems. Cheung *et al.* [20] propose a "destination set grouping (DSG)" scheme to overcome the drawback of the "listen to the slowest request" approach. In their scheme, the receivers could be divided into groups based on their capabilities, and the source carries out as many simultaneous independent connections as the number of groups. Within each group, they use window-based flow control based on the "listen to the slowest request" policy. They show that this grouping approach can improve the performance of window-controlled multipoint connections in terms of average power (throughput/delay) of the system. They propose a static grouping heuristic, in which case the resulting groups are fixed through the connection. This is a feasible solution in the cases where the performance gain via grouping can justify the cost of using multiple simultaneous multicast connections. They also propose a dynamic grouping protocol where the receivers change group as their capabilities change. This may incur too much processing overhead, and its applicable situations might be limited.

All of the above-referenced work addressing the multicast flow control issue indicates that it is a complicated and challenging problem. It is widely acknowledged that multicast flow control is neither well understood nor well studied. We are not aware of any systematic approach to analyze a class of multicast flow control algorithms as in the point-to-point case where, for example, deterministic delay-differential equations are used to model and analyze a class of unicast feedback control algorithms.

Based on the above observation of the current work, we feel that it is necessary to formulate the problem within a simple framework in order to discover the fundamental and distinctive features of multicast flow control. We focus on how to control the source transmission rate in this paper. We do not discuss how to handle the feedback implosion problem and multicast error control which have been handled recently in a number of papers [11].

Our work is distinguished from previous work in several ways. We focus on a fundamental theory for controlling source rate, with the goals of identifying the *pros* and *cons* of the "listen to the slowest request" approach and proposing better strategies. We focus on rate-based control algorithms which are easier to analyze using the fluid-flow technique. Also, [5] indicated that rate-based schemes are more suitable for multicast systems than the window-based schemes. In addition, as far as performance analysis is concerned, window-based algorithms can be analyzed by translating to a rate-based problem [21]. We consider a dynamical environment where the receiver throughput capability is changing randomly. This is typical in mixed wired/wireless networks, and we believe that it is important to include this in the model. The previous work on multicasting to mobile hosts is mainly concerned with routing messages.

Considering the complexity of the problem, and as a first attempt, we address simple situations with the following restrictions in this paper.

1) We consider data traffic only since flow control is most important for burst data traffic.
2) We consider one hop between the source and the destinations only.
3) We mainly address the case with two receivers because it is the simplest case which captures the basis of multicast connections.
4) The source uses binary on–off control, i.e., the source can only turn on and off without any intermediate transmission rate. This makes the system easier to analyze.

The insights gained from these simple situations will be helpful for studying more general cases.

The paper is organized as follows. In the next section, we propose a multicast flow control framework and discuss its essential elements. We describe three ad hoc algorithms in Section III. A special case of zero propagation delay, which manifests the effect of the phase differences of the instantaneous throughput capabilities of the receivers, is studied and analyzed using a fluid-flow analysis technique in Section IV. In Section V, the effect of propagation delay is discussed, and the performance of the two feedback control schemes proposed is compared using simulation. It turns out that the zero-delay case provides good performance bounds in the reasonable operating region for the case with nonnegligible propagation delay. Section VI concludes the paper.

II. Multicast Flow Control Framework

We propose a multicast flow control framework for data traffic traversing both a wired and wireless network. We have focused on a simple case of two mobile receivers located at different distances from the source located within the (error-free) wired network. The service rate or throughput characteristic of the mobile receivers is modeled by a two-state Markovian fluid model representing heavy fading (no throughput) and no fading (normal throughput). The Markov model for a fading process is justified in [22]. This model could also be generalized to capture the dynamics of the communication links such as shared-media links and flow-controlled links [23]. The source makes binary decisions whether to start or stop transmission according to the feedback information received from the receivers. Different feedback schemes return different amounts of information about the receiver queue occupancy. Fig. 1 illustrates a simple multicast scenario, and the corresponding abstract framework is shown

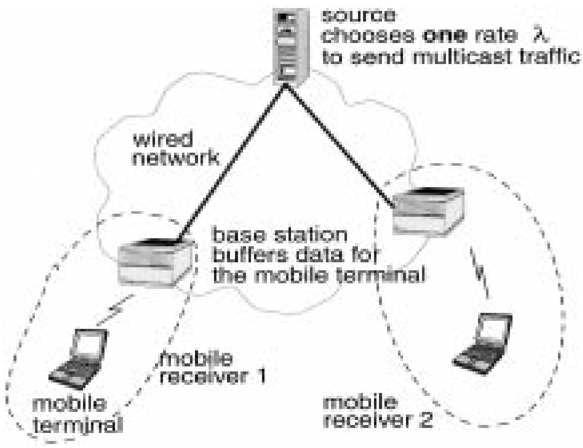


Fig. 1. Simple scenario for multicast.

in Fig. 2.[1] The on–off receivers in Fig. 2 represent a composite of the base station and wireless link of Fig. 1: the base station buffer is represented by the queue of the corresponding on–off receiver; the on–off receiver server, moving randomly between the two on–off states shown, models the fading wireless link. The feedback message is sent to the source by the base station on behalf of the mobile receivers with which the source is communicating. Therefore, we assume that the feedback messages are error free.

An essential part of a multipoint flow control algorithm is that the source has to choose a proper transmission rate based on the feedback from all the receivers. We call this decision scheme a *source policy*. The challenge is how to combine and use the feedback information from different receivers which experience different delays and could conflict with each other. Understanding the effect of different source policies is the key to the design of good multipoint flow control algorithms. We also need to specify the *feedback schemes* adopted at the receiver side.

Our framework incorporates three essential features of the real system to capture the basis of multicasting. These are the following.

1) *Stochastic receivers*, modeling the mixed wired/wireless environment;
2) *Multiple receivers*, modeling the multicast environment;
3) *Propagation delays* associated with each receiver.

Consequently, these features introduce two new phenomena of the multicast flow control problem compared to the traditional point-to-point flow control problem.

1) *Receiver phase differences* (RPD's), that is, the receivers' instantaneous throughput capabilities could be in different phases (either fading or out of fade) due to the dynamically changing available bandwidth of the multiple on–off receivers.
2) *Propagation delay differences* because the multiple receivers might be located at different distances from the source.

The effect of these features and phenomena on the design of multicast flow control algorithms can be summarized as follows.

1) The effect of receiver phase differences (RPD's) is that the source could receive conflicting feedback information from the two receivers, one requesting a speed up, the other one requesting a slow down, and vice versa. *How should the source react to the conflicting requests*?
2) The effect of propagation delay is that the feedback information obtained by the source is actually out of date. *How to handle delayed feedback information* has long been a subject under study for the unicast flow control problem.
3) The effect of propagation delay differences is that the feedback message from the receiver near the source could be newer than the one from the further out receiver. *Should the source give more weight to the newer feedback*?

Among these three questions, the effect of RPD's is the heart of the problem, and it is a new problem arising in multicasting. How to handle propagation delay and delay differences usually depends on the strategies used to handle conflicting feedback requests. For example, if the "listen to the slowest request" approach is used to extend a unicast flow control algorithm to multicast, then the effect of propagation delay is handled by the underlying unicast algorithm. In this paper, we stress the issue of RPD's and its effect on performance. The design of multicast flow control algorithms taking propagation delay differences into account is left for future work.

Obviously, the pattern of the RPD, such as the length of the period during which the two receivers are in different throughput states, directly affects the performance. Buffering can be used to accommodate RPD's and to improve throughput. An interesting question is what is the most cost-effective way to make use of the buffering in a multicast setting. The objectives of the multicast flow control algorithms are, therefore, to cope with RPD's cost effectively, in addition to the general flow control goals such as maximizing throughput and minimizing delay. These goals are correlated. Take the "listen to the slowest request" approach as an example. It handles the RPD (thus conflicting feedback request) in the most conservative way as the name suggests, i.e., it does not put any receiver in danger of overflow. To maximize throughput, the receiver should only indicate slowing down when its buffer is almost full. However, as a result, it may incur large queuing delay.

We summarize the standard notation used in the paper for a receiver on–off model in Fig. 3.

Other notation used follows: $q_i(t)$ is the buffer occupancy of receiver i at time t, B_i is the buffer size of receiver i, τ_i is the one-way propagation delay of receiver i, D_i is the round-trip propagation delay of receiver i, and $D_{\max}$ is the

[1]Note that we assume that the two mobiles are connected to different base stations. For the case where multiple mobiles in a multicast session are connected to the same base station, there are two possible solutions. 1) The base station keeps separate queues for each mobile; then the model is the same as the one we show. 2) The base station keeps one queue per multicast connection, and multicasts the traffic to the multiple destinations within its coverage area. Then the problem is that of a link layer providing multicast capability over broadcast radio, which has been studied in a number of papers. In this case, we assume that the multiple mobile destinations in one base station are represented by the slowest one.

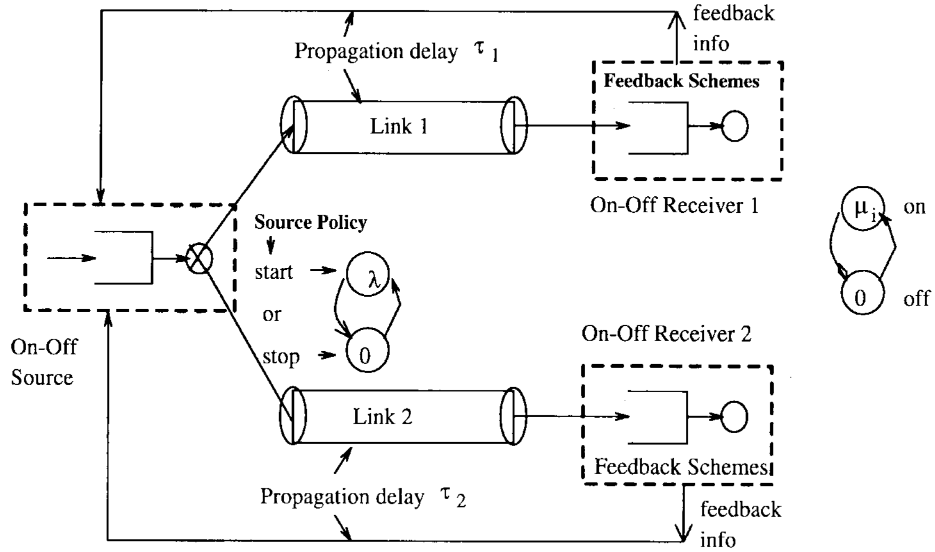


Fig. 2. Framework for multicast flow control.

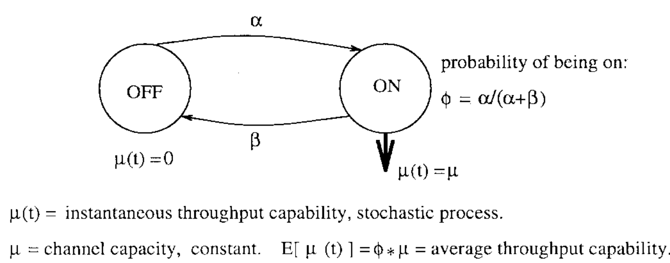


Fig. 3. Notation for a receiver on–off model.

maximum propagation delay among a group of receivers in one multicast session.

The receivers participating in a multicast connection are classified according to the following situations.

- *Homogeneous receivers*: The receivers have the same μ and ϕ. If they have the same α and β, we call them *absolutely homogeneous*.
- *Family receivers*: The receivers have the same μ, but different ϕ.
- *Heterogeneous receivers*: The receivers have different μ.

For example, if all of the active mobiles in a cellular cell are allocated one channel with the same bandwidth, we say that they have the same μ. If, furthermore, some mobiles maintain the same throughput, on the average, then they are homogeneous receivers. If they experience the same fading statistics, we call them absolutely homogeneous receivers. They are family receivers if their capabilities are different, although they are using channels with the same bandwidth.

Finally we would like to point out that the RPD phenomenon exists in all of the above three situations except for two special cases: if the receivers have the same constant throughput during the connection, then the receivers are called *in phase*; or if the receivers have different constant throughput capabilities, then the receivers are *out of phase*. In these special cases, the *stationary solution* is a constant, simply the minimum of the receiver throughput capabilities, due to the stability requirement. In the in-phase case, the "listen to the close receiver" approach could be a better solution than the "listen to the slowest request" approach. (This is one example of taking advantage of propagation delay differences.) In the out-of-phase case, the "listen to the slowest request" approach could be a good solution in most of the cases unless the transient behavior is very important, in which case some special algorithms have to be developed to meet the requirement on the transient behavior. But the assumption of constant available bandwidth at the receivers is not valid in most real systems, especially when mobile receivers are involved. Therefore, in this paper, we focus on the RPD phenomenon which makes the stationary solution of a multicast flow control algorithm nontrivial and interesting, with different control algorithms worth exploring.

III. Algorithms

A. Listen to Slowest Request (LSQ)

In this algorithm, a single-bit "start–stop" feedback scheme is adopted. That is, each receiver sends a "stop" signal to the source when its queue length crosses a high threshold, and it sends a "start" signal when the queue length drops below a low threshold. The source stops sending traffic whenever at least one receiver has sent a "stop" signal. It sends at peak rate otherwise.

To maximize throughput, we should try to have all feedback messages signal "start" as much as possible. Hence, both the high and low thresholds should be as high as possible. In our simulation, we choose the high threshold as $B_i - D_i$ for receiver i to avoid overflow [24], while the low threshold is chosen as $B_i - D_i - \Delta$ where Δ is used to avoid oscillation.

B. Source Estimation (SE)

Each receiver sends back both its queue length $q(t)$ and the queue growth rate $\dot{q}(t)$ whenever $\dot{q}(t)$ changes sign, i.e., whenever it goes from an increasing to decreasing rate, or vice versa. The source estimates the future queue length of each receiver $\hat{q}_i(t+\tau_i)$, taking the propagation delay time into account, using the feedback information. The source maintains a record of received $\dot{q}(t)$, and high and low queue threshold values for each receiver. It transmits at peak rate $\lambda = \mu$ only if at least one estimated queue length drops below its low threshold and all of the other queues stay below their high thresholds. Otherwise, it stops transmitting.

In our simulation, we choose the high threshold as $B_i - D_i$ for receiver i to avoid overflow, and the low threshold as D_i to "try" to avoid starvation [24]. We adopt a simple queue length estimation procedure taking advantage of the binary on–off control and the fact that the source transmission peak rate is equal to the receiver throughput capability during the on period, i.e., $\lambda = \mu$. Under this setting, the receiver queue size increases only if the source is on and the receiver server is off; it decreases only if the source is off and the receiver server is on; it stays the same otherwise. Based on this observation, the source updates the estimated queue length $\hat{q}_i(t + \tau_i)$ every packet length period, i.e., it increases or decreases it by one or leaves it unchanged, depending on whether the source is on or off and on the sign of the recorded queue growth rate for this receiver. Since we are using the recorded queue growth rate (which comes from a delayed feedback message) to approximate the receiver server state, the queue length estimation is not exact. The estimation error is corrected by recalculating the estimated queue length every time the source receives feedback from the receiver. The pseudocode of this estimation procedure is listed in Appendix C. We use the above estimation procedure because it is straightforward, and our simulation result shows that the estimation error is small enough for the problem at hand.

Note that for the SE and LSQ algorithms, we can avoid overflow by choosing proper high thresholds because both algorithms shut off the source if one queue exceeds the high threshold. As a result, these algorithms do not lose data (lossless service). But the algorithms cannot avoid starvation (waste of available bandwidth) as in the unicast case because the source rate is limited by the capabilities of the other receivers. This is essentially caused by the RPD phenomenon, which is a distinctive feature of multicasting.

C. Open-Loop Control

The source collects information from each receiver at call setup time as to its average rate of reception, its peak rate, and quality of service requirements, among other parameters. It then adjusts its own rate of transmission to satisfy those parameters, with no signal fed back.

In our simulation, we choose a constant peak rate λ for the source which ensures that the blocking probability is small ($<$1%). The delay–throughput performance can be easily analyzed as a special case of [25], and is independent of propagation delay.

These three algorithms are chosen for the following reasons. The open-loop algorithm is chosen to provide a basic benchmark with which to compare all strategies. The LSQ is a direct extension of the simple "start–stop" point-to-point flow control algorithm [24] to multicasting via the "listen to the slowest request" approach. We are interested to see how it performs in multicast connections and how to improve on it. It is easy to see that the LSQ copes with RPD's by letting each receiver queue store as many packets as possible to minimize the starvation caused by conflicting feedback requests from other receivers. This approach likely results in a large queuing delay. The SE algorithm is designed as a multicast-aware algorithm to be compared with the LSQ, given the lossless requirement and the constraint that the source can only turn on or off. In SE, the source maintains roughly correct information about each receiver queue size, and it adjusts its policy to try to keep the receiver queue sizes as low as possible under the constraints of minimizing starvation and avoiding buffer overflow. Therefore, the SE algorithm utilizes the receiver buffering more efficiently (resulting in less delay) at the expense of larger feedback and processing load. More accurate statements about these observations are formally presented as the properties of these algorithms in our analysis of the zero-delay case in Section IV. Our intention is to show that there are multicast-aware algorithms achieving better delay–throughput performance than the LSQ and deserving further study. The SE serves this role.

Analysis of the proposed feedback algorithms with propagation delay in a mixed wired/wireless environment, even for the case of only two mobile receivers undergoing fading, appears extremely complex since it results in coupled stochastic delay-differential equations. (The receiver random on–off link speed characteristic, which is used to model fading, results in a stochastic forcing function driving the delay-differential equations one obtains.) Therefore, initially, we have carried out extensive simulations to study the performance tradeoffs of these algorithms [26]. We have focused on the case of homogeneous receivers. We have studied the performance in terms of the throughput–delay characteristics, delay jitter, and control overhead for various system parameters. The effects of different on–off time scales of the receivers (fading statistics), buffer size, various propagation delays, and delay differences of the receivers have been studied. This experience led us to a better understanding of the algorithms, in particular, the special Markovian structures of the LSQ and SE algorithms with zero propagation delay, which can be analyzed using the techniques developed by Mitra [25].

Interestingly, the zero-delay analysis confirms most of the simulation conclusions in [26], and provides some useful interpretation, particularly as to the role played by multicast. In addition, the analysis shows that the LSQ and SE algorithms best apply to the case of homogeneous receivers, which was the focus of our simulation. In the next section, we present the analysis of the LSQ and SE algorithms with zero delay. The effect of propagation delay is discussed later.

IV. Multicast with Zero Delay

In this section, we focus on a special case: multicast without delay. It is of great importance for several reasons. It provides the performance upper bounds for the cases with nonnegligible delays. It is not a trivial problem as in the unicast case, and reveals the complexity of the problem caused by multiple receivers. It provides helpful insight into a major distinction of multicast connections: RPD's.

Consider the framework in Fig. 2 with zero propagation delays. We know that a buffer can be provided at the receiver side to accommodate the phase difference and to improve the throughput. Given that the objectives of the source policy are to maximize throughput, minimize delay at each receiver side, and to guarantee certain fairness, etc., what is the best way to utilize the receiver buffers? It is a complicated stochastic optimization problem, depending on the RPD pattern, fairness criterion, and other control objectives.

As in the unicast case, some ad hoc flow control algorithms have been selected since they are comparatively simple to analyze and work well in many situations [24]. Hence, we focus on the algorithms proposed in Section III in the zero-delay setting. The performance of the algorithms is analyzed using fluid-flow analysis as a special case of [25]. The analytical results are verified by simulation. The tradeoffs of the different algorithms and the effect of receiver phase difference are investigated.

The rest of this section is focused on a multicast session with two receivers with the same normalized $\mu = 1$. We also assume the source is persistent, i.e., it always has traffic to send if allowed. We discuss how to generalize the results to heterogeneous and/or larger numbers of receivers later.

A. LSQ with Zero Delay

The LSQ algorithm in the zero-delay case can be summarized in one sentence: the source shuts off whenever one of the receivers' queues tends to overflow. It is a simple algorithm with the following properties to be shown later.

1) It is a lossless system, and achieves maximum throughput for a given buffer size.
2) It is the simplest scheme to achieve property 1).
3) Buffer occupancy evolves periodically with alternation of one queue staying full and the other queue staying below a full buffer until they exchange position.

A formal description of the algorithm is as follows:

$$\dot{q}_i(t) = \begin{cases} \lambda(t) - \mu_i(t) & \text{if } 0 < q_i(t) < B_i, i = 1, 2 \\ [\lambda(t) - \mu_i(t)]^+ & \text{if } q_i(t) = 0, i = 1, 2 \\ [\lambda(t) - \mu_i(t)]^- & \text{if } q_i(t) = B_i, i = 1, 2 \end{cases} \quad (1)$$

$$\lambda(t) = \begin{cases} 0 & \text{if } (q_1(t) = B_1 \text{ and } \mu_1(t) = 0) \\ & \text{or } (q_2(t) = B_2 \text{ and } \mu_2(t) = 0) \\ \mu = 1 & \text{otherwise.} \end{cases} \quad (2)$$

Here, $q_i(t)$ is the queue length at time t for queue i, $\dot{q}_i(t)$ is the queue growth rate at time t, $\mu_i(t)$ is the instantaneous throughput of the on–off receiver model (Fig. 3), $\lambda(t)$ is the rate with which the source sends traffic at time t, B_i is the buffer size for receiver queue i, and $[x]^+ := \max(x, 0)$, $[x]^- := \min(x, 0)$. Equation (2) follows from the LSQ algorithm described in Section III-A by setting both the high and low thresholds to B_i for each receiver. (For the fluid-flow analysis to follow, it is valid to have $\Delta = 0$.) Note that, although the $\mu_i(t)$'s appear in the condition in (2), the information comes from the feedback on the queue size, and no additional information is assumed. This is because receiver i sends a "stop"$(= 0)$ only if its queue size tends to exceed its high threshold (B_i), which happens only if $(q_i(t) = B_i$ and $\mu_i(t) = 0)$; otherwise, if $\mu_i(t) = 1$ and $\lambda(t) = 0$, the receiver queue will drop below B_i (which is also the lower threshold) to result in a "start" message to be sent.

The lossless property is obvious. Maximum throughput for a given buffer size for a lossless system can be achieved by maximizing the ON time of the source, which in turn can be achieved by turning the source off only if it is necessary, i.e., if there is one queue tending to overflow. Note this is exactly the LSQ algorithm. It is an obvious way to achieve the maximum throughput given the lossless constraint, but results in large queuing delay because it always tries to keep the receiver buffer full.

The LSQ scheme is simple because it only requires the receiver to send a single bit of feedback whenever its queue tends to overflow or drop below a full buffer, and the source makes a decision by a logical AND operation whenever there is a feedback signal coming in. These are the simplest feedback and decision-making schemes.

Property 3) is not difficult to see from (1) and (2). Fig. 4 illustrates a typical periodic evolution of the buffer occupancy of the two receiver queues. The shaded areas indicate that $q_i(t)$ could be anywhere between $[0, B_i)$ when $q_j(t) = B_j$, depending on the outcome of the random receiver models. The solid line illustrates a possible sample path.

An interesting observation from this property is that the LSQ algorithm in the zero-delay case forces $\lambda(t) = \mu_i(t)$ when $q_i(t)$ stays full. Notice that it says exactly "*listen to the slowest receiver*" because, while queue 1 stays full and queue 2 is below a full buffer, which means that during this period receiver 1 has a lower capability than receiver 2, the source listens to receiver 1. This confirms our argument in the Introduction that the "listen to the slowest request" approach results in a throughput which takes the lower profiles of the receiver instantaneous throughput curves in a dynamical setting.

Another benefit of this result is that, during the period in which $q_1(t) = B_1$, the growth rate of queue 2 is $\mu_1(t) - \mu_2(t)$ where $\mu_1(t)$ and $\mu_2(t)$ are independent on–off Markov processes. Thus, the two queues can be decoupled except at the boundary points, and the stationary distribution of each queue can be studied as a special case of [25], where the buffer distribution of a queuing system with on–off MMF arrivals and an on–off MMF server is completely specified. By applying the fluid-flow analysis technique and manipulating the boundary conditions carefully, we solve the stationary buffer distributions for the homogeneous and family receivers. The results are stated below, and the derivation of the results is presented in Appendix A.

Case 1: The LSQ algorithm, zero propagation delay, homogeneous receivers with $\mu_1 = \mu_2 = 1$ and $\phi_1 = \phi_2 := \phi$. For $i, j \in \{1, 2\}$, we have

$$\Pr(q_i \le x) = \frac{x + b_i}{B_1 + B_2 + b_1 + b_2}, \qquad 0 \le x < B_i \quad (3)$$

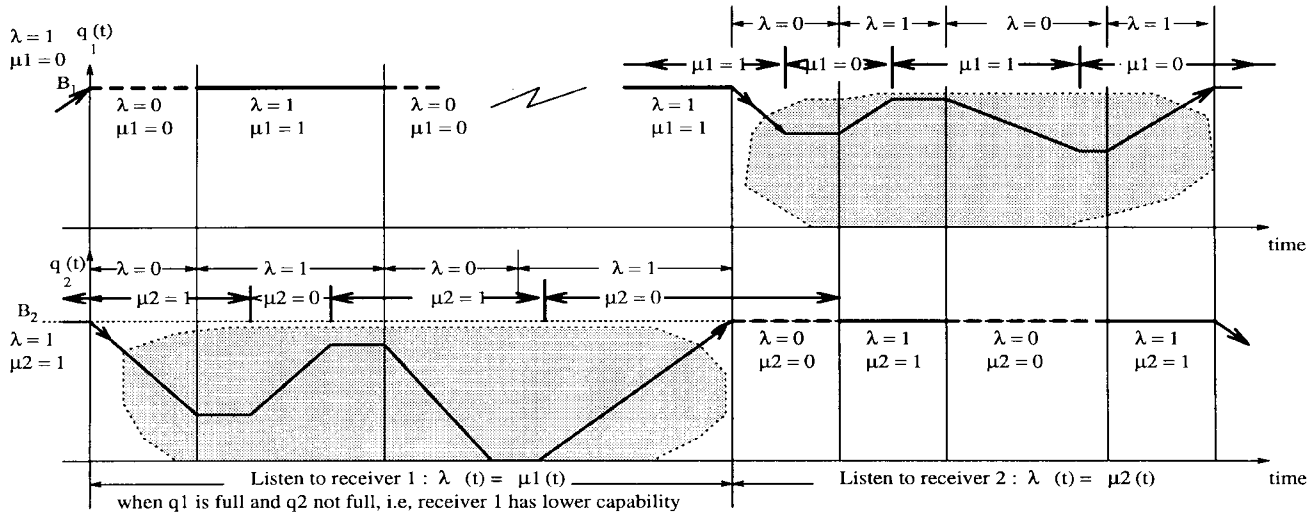


Fig. 4. Periodic evolution of the buffer occupancy for LSQ with zero delay.

$$\Pr(q_i = 0) = \frac{b_i}{B_1 + B_2 + b_1 + b_2} \tag{4}$$

$$\Pr(q_i = B_i) = \frac{B_j + b_j}{B_1 + B_2 + b_1 + b_2} \tag{5}$$

where

$$b_i = (1-\phi)\left[\frac{\phi}{\beta_i} + \frac{(1-\phi)}{\beta_j}\right], \qquad j \neq i. \tag{6}$$

The throughput[2] of the two receivers is the same and is given by

$$\gamma_i = \phi - \frac{\phi(1-\phi)(b_1+b_2)}{B_1 + B_2 + b_1 + b_2}. \tag{7}$$

The average buffer occupancy is given by

$$E\{q_i\} = \frac{1}{2} B_i \left(\frac{B_i + 2(B_j + b_j)}{B_1 + B_2 + b_1 + b_2}\right). \tag{8}$$

Case 2: The LSQ algorithm, zero propagation delay, family receivers with $\mu_1 = \mu_2 = 1$ and $\phi_1 \neq \phi_2$. For $i, j \in \{1, 2\}$, we have

[2] Long-term time average of the number of packets served by the receiver server.

$$\Pr(q_i \le x) = \pi^{(i)}(x) = \frac{1 - \dfrac{(\alpha_1+\alpha_2+\beta_1+\beta_2)^2}{(\alpha_1+\alpha_2)(\beta_1+\beta_2)}(1-\phi_i)\phi_j e^{z_i x}}{1 - \dfrac{(1-\phi_i)\phi_j}{\phi_i(1-\phi_j)} e^{z_i(B_1+B_2)}},$$
$$0 \le x < B_i,\ j \neq i \tag{9}$$

$$\Pr(q_i = 0) = \pi^{(i)}(0), \quad \Pr(q_i = B_i) = \pi^{(j)}(B_j) \tag{10}$$

where

$$z_i = \frac{(\alpha_j\beta_i - \alpha_i\beta_j)(\alpha_1+\alpha_2+\beta_1+\beta_2)}{(\alpha_1+\alpha_2)(\beta_1+\beta_2)}. \tag{11}$$

The throughput of the two receivers is given by

$$\gamma_i = \phi_i - \frac{\phi_i - \phi_j}{1 - \dfrac{(1-\phi_i)\phi_j}{\phi_i(1-\phi_j)} e^{z_i(B_1+B_2)}}. \tag{12}$$

The average buffer occupancy is given as shown in (13) at the bottom of the page.

The results will be plotted and discussed later.

B. SE with Zero Delay

The SE with zero delay case degenerates to the case for which the source is on *iff* one queue stays empty and the other queue is not full; the source is off otherwise. The SE algorithm tries to drive the two queues to zero as much as possible (least backlog) under the lossless constraint.

More precisely, with the same governing equation (1) for buffer occupancy as in the LSQ case, the control policy is

$$E\{q_i\} = \frac{B_i + \dfrac{(\alpha_1+\alpha_2+\beta_1+\beta_2)\alpha_i\beta_j}{(\alpha_1+\beta_1)(\alpha_2+\beta_2)(\alpha_j\beta_i - \alpha_i\beta_j)} e^{-z_iB_j}(e^{-z_iB_i} - 1)}{1 - \dfrac{\alpha_i\beta_j}{\alpha_j\beta_i} e^{-z_i(B_1+B_2)}}. \tag{13}$$

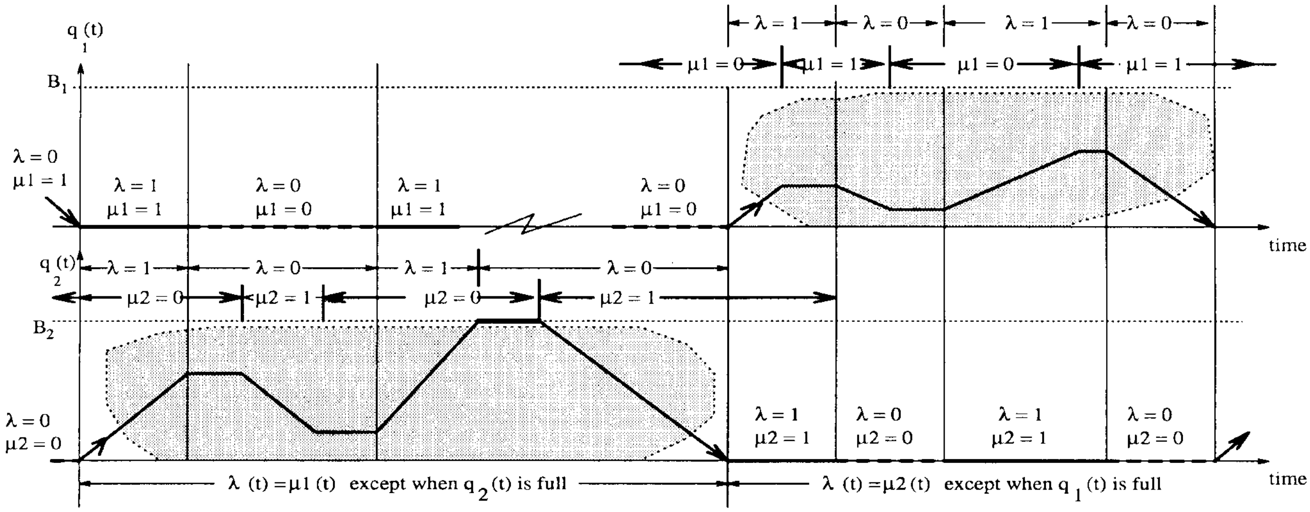


Fig. 5. Periodic evolution of the buffer occupancy for SE with zero delay.

different and is specified as follows:

$$\lambda(t) = \begin{cases} \mu = 1 & \text{if } (q_1(t) = 0 \text{ and } q_2(t) < B_2) \\ & \text{or } (q_2(t) = 0 \text{ and } q_1(t) < B_1) \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

With zero delay, the source knows $q_i(t)$ exactly because of the timely feedback of $\dot{q}_i(t)$ and $q_i(t)$. The corresponding low and high thresholds are chosen as 0 and B_i. As we will explain later, the problem is found to be a *dual problem* of the LSQ with zero delay. We will show the following properties of the SE with zero delay.

1) It is a lossless system, and achieves maximum throughput for a given buffer size.
2) It achieves minimum delay for a system with property 1).
3) The buffer occupancy evolves periodically with alternation of one queue staying empty and the other queue nonempty until they exchange position.

The periodic evolution of buffer occupancy is pictured in Fig. 5. As we can see, the figure is quite symmetric to the one for the LSQ case (Fig. 4) with the system operating at the zero boundary in the SE algorithm and the B_i boundary in the LSQ algorithm. Note that in the SE case, $\lambda(t) = \mu_i(t)$ when $q_i(t) = 0$, except that it is forced to be zero if the other queue becomes full. As in the LSQ case, the stationary buffer distribution can be solved analytically, and the results are listed below (see Appendix B for the derivation).

Case 3: The SE algorithm, zero propagation delay, homogeneous receivers, with $\mu_1 = \mu_2 = 1$ and $\phi_1 = \phi_2 := \phi$. For $i, j \in \{1, 2\}$, we have

$$\Pr(q_i \le x) = \frac{x + B_j + b_j}{B_1 + B_2 + b_1 + b_2}, \qquad 0 < x < B_i, \, j \ne i \quad (15)$$

$$\Pr(q_i = 0) = \frac{B_j + b_j}{B_1 + B_2 + b_1 + b_2} \quad (16)$$

$$\Pr(q_i = B_i) = \frac{b_i}{B_1 + B_2 + b_1 + b_2} \quad (17)$$

where b_i is given by (6).

The throughput of the two receivers is the same. Moreover, it is the same as in the LSQ case given by (7).

The average buffer occupancy is given by

$$E\{q_i\} = \frac{1}{2} B_i \left(\frac{B_i + 2b_i}{B_1 + B_2 + b_1 + b_2} \right). \quad (18)$$

Case 4: The SE algorithm, zero propagation delay, family receivers, with $\mu_1 = \mu_2 = 1$ and $\phi_1 \ne \phi_2$. For $i, j \in \{1, 2\}$, we have

$$\begin{aligned} \Pr(q_i \le x) &= \pi^{(i)}(x) \\ &= \frac{1 - \dfrac{(\alpha_1 + \alpha_2 + \beta_1 + \beta_2)^2}{(\alpha_1 + \alpha_2)(\beta_1 + \beta_2)} \phi_j (1 - \phi_i) e^{z_i (x + B_j)}}{1 - \dfrac{\phi_j (1 - \phi_i)}{\phi_i (1 - \phi_j)} e^{z_i (B_1 + B_2)}}, \\ & \qquad 0 \le x < B_i, \, j \ne i \end{aligned} \quad (19)$$

$$\Pr(q_i = 0) = \pi^{(i)}(0), \quad \Pr(q_i = B_i) = 1 - \pi^{(i)}(B_i) \quad (20)$$

where z_i is given by (11).

The throughput of the two receivers is the same as in the LSQ case given by (12).

The average buffer occupancy is given by

$$E\{q_i\} = \frac{B_i + \dfrac{(\alpha_1 + \alpha_2 + \beta_1 + \beta_2)\alpha_i \beta_j}{(\alpha_1 + \beta_1)(\alpha_2 + \beta_2)(\alpha_j \beta_i - \alpha_i \beta_j)} (e^{-z_i B_i} - 1)}{1 - \dfrac{\alpha_i \beta_j}{\alpha_j \beta_i} e^{-z_i (B_1 + B_2)}}. \quad (21)$$

Property 1) is proved by analysis which shows that the SE algorithm achieves the same throughput as the LSQ algorithm. Property 2) follows from the nature of the SE algorithm, i.e., keeping the least backlog in the buffer. The duality refers to the phenomenon that the LSQ algorithm operates at the full buffer boundary, while the SE algorithm operates at the empty buffer boundary in a symmetric way. It is evident from several aspects. The buffer occupancy evolutions are quite symmetric; the stationary buffer occupancy distributions show that $\Pr(q_i = B_i)$ and $\Pr(q_i = 0)$ in the LSQ case are equal to $\Pr(q_i = 0)$ and $\Pr(q_i = B_i)$ in the SE case, respectively. The derivations of the stationary distributions described in the Appendix precisely reflect the duality of the two systems.

C. Comparison

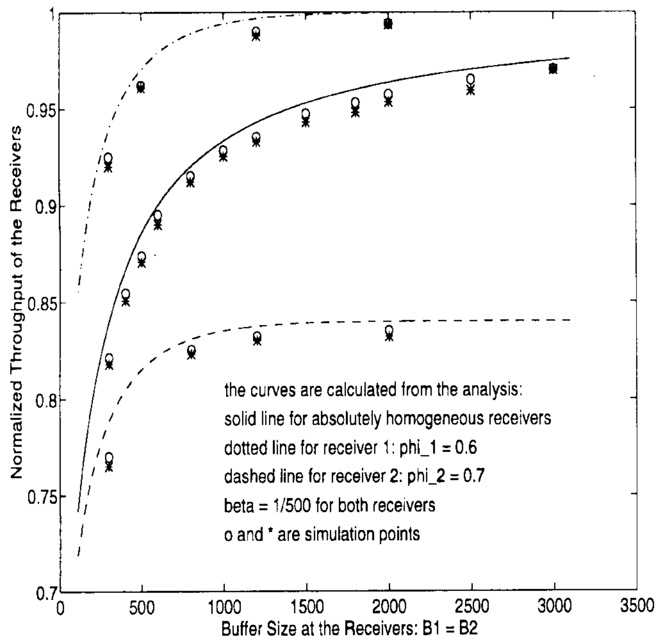
The fluid analysis directly applies to the case of two heterogeneous receivers as well, as discussed in the Appendix, but there is no closed-form solution. In general, the analysis of a multicast connection with n receivers requires solving n correlated buffer distributions. Property 3) of the LSQ and SE algorithms in the zero-delay case reduces the dimension of the problem to $n-1$, and thus the two-receiver case reduces to a special case of [25]. A higher dimensional system with $n \geq 3$ is very difficult to solve to obtain useful results.

The closed-form solutions listed above can be used to study the effect of system parameters on the throughput–delay performance. The results are verified by simulation. Some of the results we have obtained are summarized below.

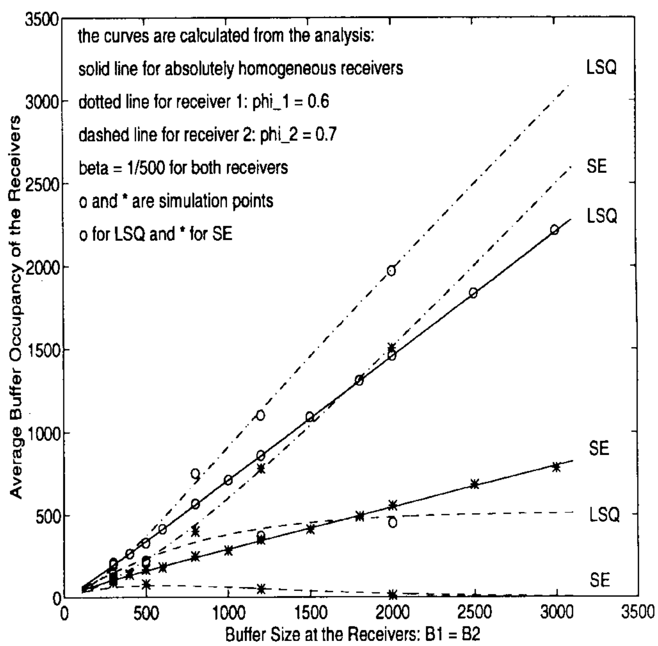
In our study, we choose a receiver 1 with $\mu_1 = 1$, $\beta_1 = 1/500$, and $\phi_1 = 0.6$ (cases with other values of β and ϕ exhibit the same trend). All time units are in packet lengths. We compare the performance when receiver 1 participates in a multicast session with different kinds of receivers. In Figs. 6 and 7,[3] the curves are computed from analysis with simulation points superimposed. [In Fig. 7(b), we only draw the simulation points corresponding to the absolutely homogeneous receiver case to keep the picture clear.] We can see that the analytical results fit the simulation results well. The slight discrepancy is due to the fluid approximation which gives a larger throughput than the simulation results.

Fig. 6 compares the performance of the family receivers with that of the absolutely homogeneous receivers. For the family receiver case, we picked another receiver 2 with $\mu_2 = 1$, $\beta_2 = 1/500$, and $\phi_2 = 0.7 > \phi_1$. The results show that for family receivers with different throughput capabilities, the faster receiver 2 is limited by the slower one and suffers from reduced throughput. More importantly, if we apply these two algorithms to the family receiver case, the slower receiver experiences too large a delay, even for the SE algorithm [see Fig. 6(b)]. Therefore, we suggest avoiding the use of the LSQ and SE algorithms to multicasting with receivers of different capabilities. We should group the homogeneous receivers together, and set up multicast connections based on such groups whenever possible. Cheung *et al.* [20] suggested the same idea, and provided algorithms to realize such grouping; their other work [27], [28] suggested that such grouping can also improve fairness and throughput of data-link layer protocols.

Fig. 7 illustrates the effect of different on–off time scales for homogeneous receivers. Fig. 7(b) shows that there is no significant difference in the delay performance with different on–off times. Receiver 1 achieves better throughput if it is in a session with the other receiver having faster on–off time scales; the throughput reduces otherwise. The throughput reduction is significant only if the differences of the on–off times are huge, and it vanishes as the buffer size increases. Therefore, we conclude that the case of absolutely homogeneous receivers is of the greatest interest to study because the LSQ and SE algorithms are best applied to the homogeneous receiver case, and small variations of on–off time scales among homogeneous receivers do not cause too much difference in the throughput–delay performance.



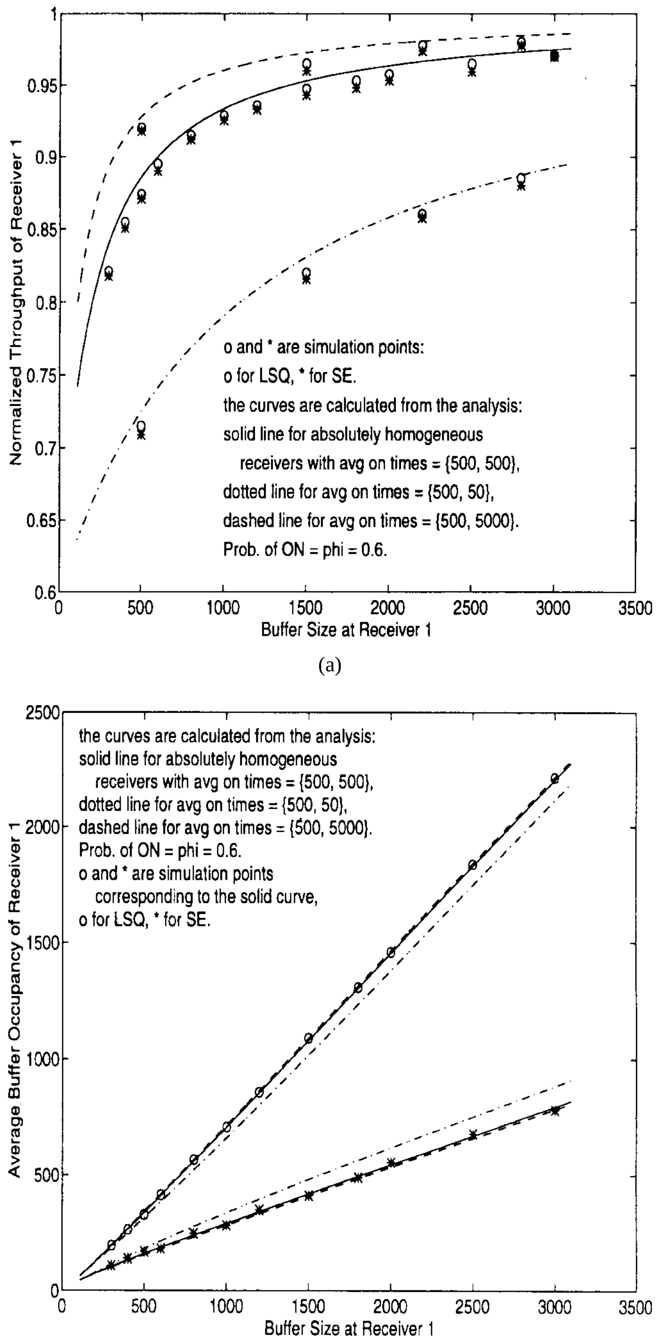
(a)



(b)

Fig. 6. Performance comparison: homogeneous versus family receivers. (a) Throughput. (b) Average buffer occupancy.

[3] The normalized throughput shown in the figures is calculated as the long-term time average of the number of packets served by the receiver server (definition of throughput) divided by ϕ.



(b)

Fig. 7. Performance comparison: homogeneous receivers. (a) Throughput. (b) Average buffer occupancy.

Comparing the LSQ with the SE algorithm, they achieve the same throughput in this zero-delay case, but the SE algorithm has a better delay performance. We have drawn a similar conclusion based on our earlier simulation results in [26] where we observed similar throughput performance of the two algorithms and better delay performance of the SE algorithm in the presence of propagation delay. For the zero-delay case with homogeneous receivers, we can show from the analysis that the ratio of the two mean buffer occupancies is $E(q)^{\mathrm{LSQ}}/E(q)^{\mathrm{SE}} = (3B+2b)/(B+2b)$, which is roughly three if buffer size B is large compared to $b = (1-\phi)/\beta$ which is on the order of the average on time $1/\beta$. A ratio of three represents a significant reduction of the average delay in the SE algorithm.

V. Effect of Propagation Delay

The effect of propagation delay in a multicast connection can be summarized as follows.

1) The propagation delay differences contribute to the receiver phase differences.
2) The propagation delay affects the control capability of the source policy.
3) The presence of the propagation delay makes the system non-Markovian, and the fluid-flow analysis cannot be applied directly.

With the presence of propagation delays, the feedback signals experience different delays from different receivers. Then the source acquires an inaccurate picture of the receivers' status through feedback in which the phase differences of the receivers' status could be due to either the phase differences of the receiver capability or the delay differences among the receivers. Therefore, the source faces a tougher challenge of how to react to the different requests from the receivers. Using fluid models, such a system can be formulated as a set of stochastic delay-differential equations which, at this point, appear too difficult to solve to obtain any useful results. Approximations could be used to restore the Markovian property in order to apply the fluid-flow analysis. This work is proceeding.

We discuss next how the LSQ and SE algorithms cope with delay and the effect of delay, on the performance degradation.

A. LSQ and SE with Delay

With the presence of propagation delays, the algorithms have to adjust the threshold values to avoid overflow and starvation, as discussed in Section III. This results in less throughput and larger queuing delay in general.

The effect of propagation delay on the LSQ algorithm is simple. The source shuts off more than necessary because the high threshold $B_i - D_i$ is a worse case consideration to avoid any possibility of buffer overflow.

For the SE algorithm, the presence of the propagation delay makes control more difficult. We need to estimate the future queue length in the SE algorithm. There are estimation errors, and the accuracy depends on the estimation procedure and how much information the source has. The estimation procedure may require intensive processing, and the estimation error may further reduce the throughput. Thus, the propagation delay has a more severe impact on the SE algorithm than the LSQ algorithm.

B. Simulation Results

Here, we present some simulation results to illustrate the performance degradation due to the presence of propagation delay, and we also contrast the performance of different algorithms. The parameters used are for absolutely homogeneous receivers with $\beta = 1/120$, $\alpha = 1/100$, $B_1 = B_2$ varying from $2D_{\max}$ to 3000, and $\Delta = 10$. The propagation delays D_1

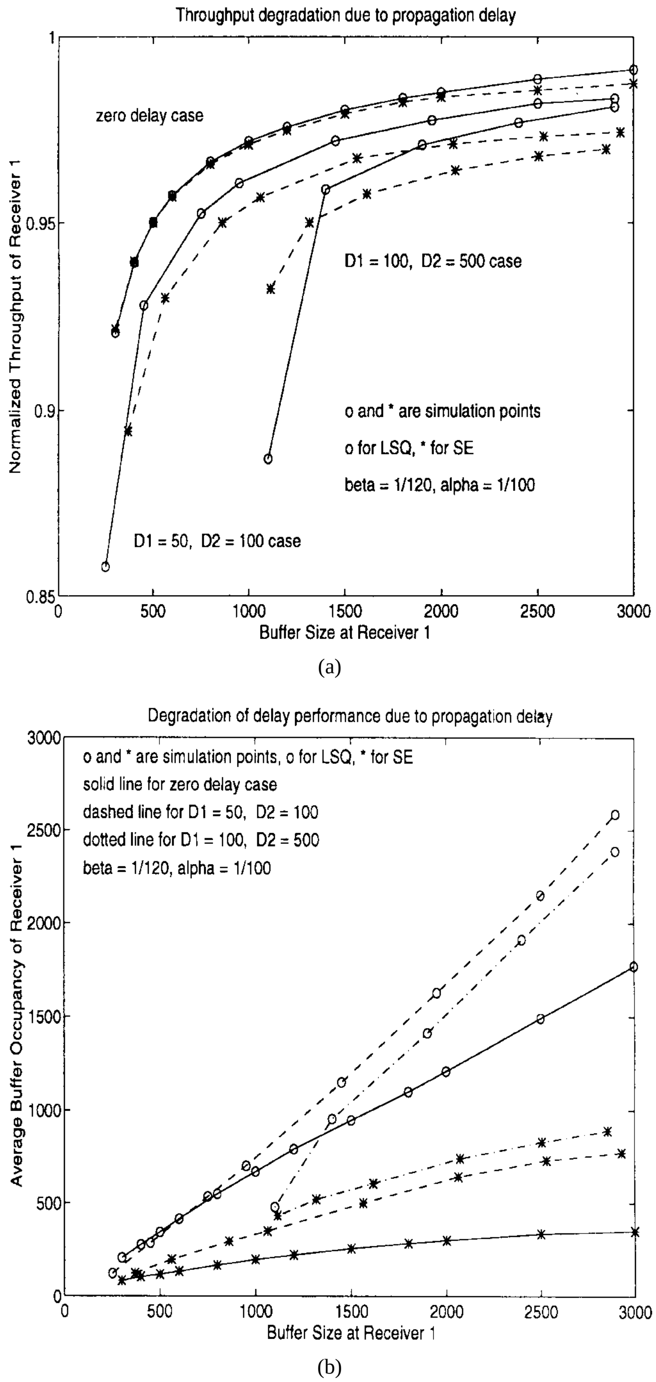


Fig. 8. Performance degradation due to propagation delay. (a) Throughput. (b) Average buffer occupancy.

and D_2 are as indicated in Fig. 8. Note that the set $\{D_1 =$ 100, $D_2 = 500\}$ has a much larger delay and a larger delay difference than the set $\{D_1 = 50, D_2 = 100\}$. Note again that time units are in packet lengths. Fig. 8 describes the performance of receiver 1, the one closer to the source.

As we can see from Fig. 8(a), the throughput reduces as the propagation delay increases. The results are as expected. The source is shut off when one of the buffer occupancies is bigger than its $B-D$, which results in a large throughput drop. The reduction is not very significant if the buffer is reasonably large, and it vanishes as the buffer size tends to infinity. The throughput also reduces more for the SE algorithm than for the LSQ algorithm, as expected.

Fig. 8(b) shows that the average buffer occupancy (and thus average queuing delay) increases if the propagation delay is present. The large increment of mean buffer occupancy in the LSQ strategy is mainly because of the delay difference of the two receivers which makes the one further out, number 2, a "slower" one. The LSQ algorithm favors the one further out, which results in larger buffer occupancy at receiver 1, as shown in the figure. If we draw the mean buffer occupancy of receiver 2, it is actually below the curve for the zero-delay case. Note also that the curve for $\{D_1 = 50$ and $D_2 = 100\}$ is actually below the one for $\{D_1 = 100$ and $D_2 = 500\}$. This is because the high threshold is chosen as $B_i - D_i$, which is lower for the latter case. On the other hand, for the SE algorithm, the mean buffer occupancy increases as the propagation delay increases because the lower threshold is chosen as D_i.

Examining Fig. 8(a) and (b) together, we can see a reasonable operating region for the LSQ algorithm is $B_1 \in (500, 1000)$. The mean buffer occupancy grows too much beyond this region. The reasonable operating region for SE is $B_1 \in (1000, 2000)$ which achieves larger throughput but with smaller average delay at the expense of more processing and more signaling load compared to the LSQ algorithm. The throughput gain is small when $B_1 > 2000$.

As we can see from the figures, the zero-delay case exhibits the same trend as the cases with propagation delay present. The zero-delay case provides a useful bound for the throughput–delay performance in the reasonable operating region.

C. Summary

The analysis of multicast without delay allows us to compare how different algorithms handle RPD with the presence of multiple receivers. The results show that, with zero propagation delay, the SE algorithm achieves better delay performance than the LSQ with the same throughput. It is also easy to see that the performance degenerates with the presence of propagation delay, as shown in the previous section. Noticing that the performance of the SE might degrade more as propagation delays become larger, it is desirable to study the performance of the algorithms under different combinations of propagation delays and receiver on–off time scales, to see where the SE has the best improvement over the LSQ to justify its cost of larger feedback and processing load.

With a lack of effective analytical tools for now, we have carried out extensive simulation to study the performance tradeoffs of the three proposed control algorithms under different system parameters, with focus on the effect of different receiver on–off time scales (fading statistics), different propagation delays, and delay differences. The detailed results are presented in our earlier paper [26], and are not included here due to space limitations. We summarize the observations obtained there in the following discussion.

The simplicity of the LSQ algorithm makes it a very attractive solution in the cases where bandwidth and processing capacity are at a premium, such as in PCS and wireless networks. Our simulation shows that, for homogeneous receivers, the LSQ algorithm works well when the receivers' fading durations are relatively small compared to the propagation delay of the receiver closest to the source. When the fading duration becomes larger than the propagation delay, the throughput of the LSQ degenerates. The LSQ cannot maintain a good throughput with a reasonable queuing delay if the receiver fading durations are very long. The SE is better in these situations because it achieves reasonable delay, even for a relatively large buffer size. Our simulation results showed that the performance improvement of the SE over LSQ reduces as the propagation delay or delay difference gets larger. Therefore, the LSQ might be favorable in the cases with large propagation delay or delay difference. To summarize, the SE best applies to both delay- and throughput-critical applications (for which the performance of the LSQ is not adequate) in the systems with larger receiver on–off time scales (slow fading), larger buffer size, but not too large propagation delays. It is well suited to the case of a huge amount of data transfer from a high-performance server to multiple receivers.

Our simulations also showed that the two feedback schemes outperform the open-loop algorithm in terms of throughput, assuming that the packet loss rate has to be low. The open-loop procedure achieves delay performance similar to that of the SE algorithm. The significant drawbacks of the open-loop control are that it cannot avoid some packet loss and, more importantly, it cannot adapt to the system changing. But there may be some cases involving long propagation delays, multiple mobile receivers with rapid fading characteristics, in which feedback flow control algorithms will not be effective. In these cases, one might have no choice but to resort to open-loop control.

As we can see, the simulation approach enables us to compare the relative impact of propagation delay and receiver on–off time scales, but not in a precise manner. That is, we observed that the LSQ and SE have their relatively good operating environments, but we cannot specify the condition in terms of system parameters. Therefore, an analytical approach for systems with propagation delay is highly desirable and is currently under investigation.

VI. Conclusions and Future Work

In summary, the performance analysis and comparison of multipoint flow control algorithms over combined wireless/wired networks is a brand-new area, and represents a significant research challenge. There are three major challenges we face: multiple receivers (multicasting), stochastic receivers (combined wired/wireless networks), and propagation delay and delay differences. The buffer behaviors of multiple correlated queues governed by stochastic delay-differential equations need to be studied, but this obviously presents great difficulty.

In this paper, we address the problem via a divide-and-conquer approach. We are able to do some performance analysis in the case where we assume zero propagation delay, and select algorithms with a special property to be able to decouple the buffer evolution behavior. The effect of the propagation delay in multicast connections is then discussed. Our work shows that the multicast flow control in the simplest form is already a very difficult problem, but is of great interest. Currently, we are investigating a method to approximate, by a Markovian system, the multicast flow control problem with multiple receivers and propagation delay. In this manner, we can study more complicated algorithms with delay incorporated.

Appendix A
The Derivation of Stationary Buffer Distributions for LSQ with Zero Delay

We assume that the modulating Markov chains for the two receiver servers are independent. Then they together can be described by the composite process $\{\mu_1(t), \mu_2(t)\}$ which is a two-dimensional Markov chain on state space $\mathcal{S}_M = \{$(on, on) $= 1$, (on, off) $= 2$, (off, on) $= 3$, (off, off) $= 4\}$ with generator M. Denote the stationary distribution of this process by $\boldsymbol{p} := \{p(\text{on, on}), p(\text{on, off}), p(\text{off, on}), p(\text{off, off})\} := \{p_i\}_{i=1,2,3,4}$. Then

$$\boldsymbol{p} = \{\phi_1\phi_2, \phi_1(1-\phi_2), (1-\phi_1)\phi_2, (1-\phi_1)(1-\phi_2)\}.$$

The generator matrix M is given by

$$M = \begin{bmatrix} -(\beta_1+\beta_2) & \beta_2 & \beta_1 & 0 \\ \alpha_2 & -(\alpha_2+\beta_1) & 0 & \beta_1 \\ \alpha_1 & 0 & -(\alpha_1+\beta_2) & \beta_2 \\ 0 & \alpha_1 & \alpha_2 & -(\alpha_1+\alpha_2) \end{bmatrix}.$$

Now, we consider the distributions of the fluid content of the two receiver buffers. As we described in Section IV, the LSQ algorithm in the zero-delay case adds a constraint so that, during the period where $q_i(t) = B_i$, $\lambda(t) = \mu_i(t)$, and $q_j(t) < B_j$ for $i, j = 1, 2$ and $i \neq j$. Equation (1) becomes

$$\dot{q}_i(t) = \begin{cases} \mu_j(t) - \mu_i(t) & \text{for } i \neq j,\ 0 < q_i(t) < B_i \\ [\mu_j(t) - \mu_i(t)]^+ & \text{if } q_i(t) = 0 \\ [\mu_j(t) - \mu_i(t)]^- & \text{if } q_i(t) = B_i \text{ and } q_j(t) = B_j \\ 0 & \text{if } q_i(t) = B_i \text{ and } 0 < q_j(t) < B_j \end{cases} \tag{22}$$

where $[x]^+ := \max(x, 0)$ and $[x]^- := \min(x, 0)$.

For x_i real and nonnegative, let vector $\boldsymbol{\pi}^{(i)}(x) = \{\pi_k^{(i)}(x)\}_{k=1,2,3,4}$ where $\pi_k^{(i)}(x) = \Pr(q_i < x, q_j = B_j \text{ state is } k)$ is the stationary distribution of the buffer occupancy of receiver i when the system is in state k for $0 < x < B_i$. It was shown in [25] that $\boldsymbol{\pi}^{(i)}$ is the solution of the differential equation

$$\frac{d}{dx}\boldsymbol{\pi}^{(i)}(x)D^{(i)} = \boldsymbol{\pi}^{(i)}(x)M, \qquad 0 \le x \le B_i,\ i = 1, 2 \tag{23}$$

where the matrix $D^{(i)} = \text{diag}\{\mu_j - \mu_i, (-1)^i\mu_1, (-1)^{i+1}\mu_2, 0\}$ is diagonal and is called the drift matrix. The diagonal elements are all possible instantaneous buffer growth rates for queue $i, (i = 1, 2)$. Denote $D_k^{(i)}$ as the (k, k)th element of the matrix $D^{(i)}$.

The general solution of (23) is treated thoroughly in [25]. Here, we have different boundary conditions because of the correlation of the two receiver queues. Let d_0, d_+, d_- denote the number of zero, positive, and negative elements on the diagonal of the drift matrix $D^{(i)}$, respectively. Also $d := d_+ + d_-$. Note that they are the same for both $D^{(1)}$ and $D^{(2)}$. Denote $\pi^{(i)}(x) = \Sigma_k \pi_k^{(i)} = \Pr(q_i < x, q_j = B_j)$, and let $W_i := \pi^{(i)}(B_i) = \Pr(q_i < B_i, q_j = B_j)$, which is also equal to $\Pr(q_i \leq B_i, q_j = B_j)$ because $(q_i = B_i, q_j = B_j)$ can only happen at isolated points in time. The two receivers are correlated by $W_1 + W_2 = 1$. Note that $W_{\boldsymbol{i}}$ is the proportion of the time that queue $\boldsymbol{j}$ stays full. The marginal distribution of the buffer occupancy for receiver i is just $\Pr(q_i < x) = \pi^{(i)}(x)$ for $0 < x < B_i$, and $\Pr(q_i = 0) = \pi^{(i)}(0)$, $\Pr(q_i = B_i) = \pi^{(j)}(B_j) = W_j$.

Note that to solve (23), we need d boundary conditions for each $i (i = 1, 2)$. They are given by

$$D_k^{(i)} > 0 \Rightarrow \pi_k^{(i)}(0) = 0 \tag{24}$$

$$\pi_k^{(1)}(B_1) + \pi_k^{(2)}(B_2) = p_k, \qquad k = 1, 2, 3, 4. \tag{25}$$

Equation (24) is conventional, and it gives d_+ conditions for each i. Equation (25) captures the correlation of the two receivers; it says that in state k, either receiver 1 or receiver 2 stays full. There are $2d_-$ independent equations in (25). The throughput rate of each receiver is given by

$$\gamma_i = \mu_i \phi_i + \sum_{D^{(i)}(k) < 0} \pi_k^{(i)}(0) D^{(i)}(k). \tag{26}$$

Note that there is no loss for this system.

The dimension of the differential equation (23) is d for each i. In this paper, we focus on the case where $\mu_1 = \mu_2 = \mu$. Equation (23) then reduces to a two-dimensional problem and can be solved explicitly. Note that this is the case of multicast sessions with homogeneous receivers and family receivers which are very common and important in real systems. The case of $\mu_1 \neq \mu_2$, i.e., heterogeneous receivers, can always be solved by the numerical technique described in [25], but we cannot write the distributions in closed form.

If $\mu_1 = \mu_2 = \mu = 1$, then $d_0 = 2$, $d_+ = d_- = 1$, and (23) reduces to

$$[\dot{\pi}_2^{(i)}(x) \quad \dot{\pi}_3^{(i)}(x)] = [\pi_2^{(i)}(x) \quad \pi_3^{(i)}(x)] \boldsymbol{A}^{(i)}, \qquad 0 \leq x \leq B_i \tag{27}$$

where the matrix $\boldsymbol{A}^{(i)}$ is given by

$$\boldsymbol{A}^{(i)} = (-1)^i \Gamma \begin{bmatrix} -\alpha_2\beta_1 & -\alpha_2\beta_1 \\ \alpha_1\beta_2 & \alpha_1\beta_2 \end{bmatrix} \tag{28}$$

$$\Gamma := \frac{\alpha_1 + \alpha_2 + \beta_1 + \beta_2}{(\alpha_1 + \alpha_2)(\beta_1 + \beta_2)} \tag{29}$$

and $\pi_1^{(i)}$, $\pi_4^{(i)}$ are specified by linear combinations:

$$\pi_1^{(i)}(x) = (\alpha_2 \pi_2^{(i)}(x) + \alpha_1 \pi_3^{(i)}(x))/(\beta_1 + \beta_2) \tag{30}$$

$$\pi_4^{(i)}(x) = (\beta_1 \pi_2^{(i)}(x) + \beta_2 \pi_3^{(i)}(x))/(\alpha_1 + \alpha_2). \tag{31}$$

The boundary conditions are

$$\pi_3^{(1)}(0) = 0, \quad \pi_2^{(2)}(0) = 0 \tag{32}$$

$$\pi_k^{(1)}(B_1) + \pi_k^{(2)}(B_2) = p_k, \qquad k = 2, 3. \tag{33}$$

Matrix $\boldsymbol{A}^{(i)}$ has two eigenvalues $z_1^{(i)} = 0$ and $z_2^{(i)} = (-1)^i(\alpha_1\beta_2 - \alpha_2\beta_1)\Gamma$. We can now solve the problem for two separate cases.

Case 1: Homogeneous receivers with $\mu_1 = \mu_2 = \mu = 1$ and $\phi_1 = \phi_2 = \phi$, which implies $z_1^{(i)} = z_2^{(i)} = 0$.

Case 2: Family receivers with $\mu_1 = \mu_2 = \mu = 1$ and $\phi_1 \neq \phi_2$. Here, $z_2^{(i)} \neq 0$.

In both cases, we can explicitly solve the two-dimensional differential equations with the specified boundary conditions. The results are listed in Section IV-A.

VIII. Appendix B
The Derivation of Stationary Buffer Distributions for SE with Zero Delay

The derivation of buffer stationary distributions for SE with zero delay can be studied as a dual problem to the LSQ case. The underlying Markov chain is the same. The constraint in the SE with the zero-delay case is that $q_i(t) = 0$ in a period. This implies that $q_j(t) > 0$ and $\lambda(t) = \mu_i(t)$ if $q_j(t) < B_j$ or $\lambda(t) = 1$ if $q_j(t) = B_j$ for $i, j = 1, 2$ and $i \neq j$. The corresponding equation (14) becomes

$$\dot{q}_i(t) = \begin{cases} \mu_j(t) - \mu_i(t) & \text{for } i \neq j,\ 0 < q_i(t) < B_i \\ [\mu_j(t) - \mu_i(t)]^+ & \text{if } q_i(t) = 0 \text{ and } q_j(t) = 0 \\ [\mu_j(t) - \mu_i(t)]^- & \text{if } q_i(t) = B_i \\ 0 & \text{if } q_i(t) = 0 \text{ and } 0 < q_j(t) \leq B_j. \end{cases} \tag{34}$$

We define the complementary probability vector for SE as $\boldsymbol{\pi}^{(i)}(x) = \{\pi_k^{(i)}(x)\}_{k=1,2,3,4}$ where $\pi_k^{(i)}(x) = \Pr(q_i \geq x, q_j = 0, \text{ state is } k)$, $0 \leq x \leq B_i$. Then $\boldsymbol{\pi}^{(i)}$ also satisfies (23) with the same $D^{(i)}$ and M as in the LSQ case.

Denote $\pi^{(i)}(x) = \Sigma_k \pi_k^{(i)} = \Pr(q_i \geq x, q_j = 0)$, and let $W_i := \pi^{(i)}(0) = \Pr(q_i \geq 0, q_j = 0)$, which is exactly $\Pr(q_j = 0)$ or the proportion of the time that queue j stays empty. In the SE case, $(q_i = 0, q_j = 0)$ can only happen at isolated points in time. The two receivers are correlated by $W_1 + W_2 = 1$. Note that $\pi^{(i)}(B_i) = \Pr(q_i \geq B_i, q_j = 0) = \Pr(q_i = B_i, q_j = 0)$ because $q_i > B_i$ is impossible. The marginal distribution of the buffer occupancy for receiver i is given by $\Pr(q_i \geq x) = \pi^{(i)}(x)$ for $0 < x < B_i$, and $\Pr(q_i = 0) = W_j$, $\Pr(q_i = B_i) = \pi^{(i)}(B_i)$. We also have that $\Pr(q_i < x) = 1 - \pi^{(i)}(x)$ for $0 < x < B_i$.

The boundary conditions in this case are

$$D_k^{(i)} < 0 \Rightarrow \pi_k^{(i)}(B_i) = 0 \tag{35}$$

$$\pi_k^{(1)}(0) + \pi_k^{(2)}(0) = p_k, \qquad k = 1, 2, 3, 4. \tag{36}$$

Equation (35) holds because, when the drift is negative, it is impossible for queue i to stay full ($q_i = B_i$ can only happen at isolated points in time).

For the same reason as stated in the LSQ case, we focus on the case with $\mu_1 = \mu_2 = \mu = 1$ and solve the two-dimensional problem (27) for both the homogeneous and family receiver cases. Then the boundary conditions can be specified as follows:

$$\pi_2^{(1)}(B_1) = 0, \quad \pi_3^{(2)}(B_2) = 0 \tag{37}$$

$$\pi_k^{(1)}(0) + \pi_k^{(2)}(0) = p_k, \qquad k = 2, 3. \tag{38}$$

The normalized throughput is then given by $\gamma_i = \phi_i - \Pr(q_i = 0$, server i is on and source is off). The second term is equal to $\Pr(q_i = 0, q_j = B_j$, server i is on). For receiver 1, it is nonzero only in state (on, off), while it is nonzero only in state (off, on) for receiver 2. Therefore

$$\gamma_1 = \phi_1 - \pi_2^{(2)}(B_2) \tag{39}$$

$$\gamma_2 = \phi_2 - \pi_3^{(1)}(B_1). \tag{40}$$

We can then solve the equations explicitly to get the results presented in Section IV.

IX. Appendix C
Pseudocode of the Receiver Queue Size Estimation Procedure for the SE Algorithm

C.

For each receiver i, the source maintains a pair of high and low threshold values (H_i, L_i), and a record of $\dot{q}_i$, which is the last received queue growth rate from this receiver. $\dot{q}_i =$ INC or DEC to indicate that the receiver queue is increasing or decreasing, respectively. The source also maintains an estimation of queue size $\hat{q}_i = \hat{q}_i(t + \tau_i)$ for each receiver i. $\lambda(t) =$ ON or OFF indicates thst the source is ON or OFF, respectively. The estimation procedure has two parts.

1) The source updates $\hat{q}_i$ every packet length period according to the following procedure, and then the source policy is updated based on the estimated queue length.

```
For each receiver i:
        if (q̇_i ≡ INC and λ(t) ≡ ON)
              q̂_i = q̂_i + 1;
        else if (q̇_i ≡ DEC and λ(t) ≡ OFF
              and q̂_i > 0)
              q̂_i = q̂_i − 1;
        else q̂_i = q̂_i;
/* update source policy λ(t) */
if ((q̂_1 ≥ H_1) or (q̂_2 ≥ H_2))
        λ(t) =OFF;
else if ((q̂_2 < L_2) or (q̂_1 < L_1))
        λ(t) =ON;
else λ(t) =OFF;
/* the following is used in part 2 */
if source policy changed, record the
current time;
```

2) The source recalculates the estimated queue length when it receives a feedback message from receiver i.

```
If the feedback message (q_i(t), q̇_i(t))
is from receiver i:
        calculate the source on-times
        and off-times within the last
        τ_i interval;
        if (q̇_i ≡INC)
              q̂_i = q_i(t)+ on-times;
        else  q̂_i = q_i(t)− off-times;
        if (q̂_i < 0)   q̂_i = 0;
```

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] M. Schwartz, "Network management and control issues in multimedia wireless networks," *IEEE Personal Commun.*, pp. 8–16, June 1995.
[2] Y. Amir, D. Dolev, S. Kramer, and D. Malki, "Transis: A communication sub-system for high availability," in *Proc. 22nd IEEE Int. Symp. Fault-Tolerant Computing,* July 1992, pp. 76–84.
[3] K. P. Birman and R. V. Renesse, *Reliable Distributed Computing with the ISIS Toolkit* IEEE Computer Society Press, 1994.
[4] R. V. Renesse, T. M. Hickey, and K. P. Birman, "Design and performance of Horus: A lightweight group communications system," Tech. Rep. TR94-1442, Dep. Comput. Sci., Cornell Univ., Aug. 1994.
[5] G. D. H. Hunt, "Multicast flow control on local area networks," Ph.D. dissertation, TR95-1479, Dept. Comput. Sci., Cornell Univ., May 1995.
[6] X.-M. Chen, "Gigabit network multicast protocols," ch. 4, "Flow control techniques," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. California, Santa Barbara, 1996.
[7] A. P. Naryan, "Reliable transfer of data in a local area network with multicast distribution," in *Proc. 15th IEEE Conf. Local Comput. Networks*, Sept. 1990, pp. 310–319.
[8] B. J. Dempsey, J. C. Fenton, and A. C. Weaver, "The multidriver: A reliable multicast service using the Xpress transfer protocol," in *Proc. 15th IEEE Conf. Local Comput. Networks*, Sept. 1990, pp. 351–358.
[9] S. Armstrong, A. Freier, and K. Marzullo, "Multicast transport protocol," RFC 1301, Xerox, Apple, Cornell Univ., 1992.
[10] R. Braudes and S. Zabele, "Requirements for multicast protocols," RFC 1458, TASC, 1993.
[11] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *Proc. IEEE INFOCOM'96,* San Francisco, CA, Mar. 1996, pp. 1414–1424.
[12] R. Yavatkar, J. Griffoen, and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in *Proc. ACM Multimedia*, San Francisco, CA, 1995, pp. 333–344.
[13] GlobalCast Comm. Inc., "Reliable multicast transport (RMP)," http://research.ivv.nasa.gov/RMP/Docs/index.html.
[14] ATM Forum, "Traffic management specification version 4.0," ATM Forum/95-0013R11, Mar. 1996.
[15] D. Cavendish, S. Mascolo, and M. Gerla, "Rate based congestion control for multicast ABR traffic," in *Proc. GLOBECOM'96,* London, England, Nov. 1996, pp. 1114–1118.
[16] A. Acharya and B. R. Badrinath, "A framework for delivering multicast messages in networks with mobile hosts," *ACM/Baltzer J. Wireless Networks* (*Special Issue on Routing in Mobile Communication Networks*), to be published.
[17] B. Sabata, M. J. Brown, and B. A. Denny, "Transport protocols for reliable multicast: TRM," in *Proc. IASTED Int. Conf. Networks,* Orlando, FL, Jan. 1996.
[18] S. Pejhan, M. Schwartz, and D. Anastassiou, "Refinements to rate-based congestion control with extensions to multicast, multimedia applications," in *Proc. IFIP/IEEE Broadband Commun. '96,* Montreal, Canada, 1996, pp. 147–160.
[19] J. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *Proc. ACM SIGCOMM'94,* San Francisco, CA, 1994, pp. 58–67.

[20] S. Y. Cheung and M. H. Ammar, "Using destination set grouping to improve the performance of window-controlled multipoint connections," in *Proc. Int. Conf. Comput. Commun. and Networks*, Las Vegas, NV, Sept. 1995, pp. 388–395.

[21] M. Schwartz, *Broadband Integrated Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1996.

[22] H.-S. Wang and P.-C. Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 353–357, May 1996.

[23] K. Lee, "Performance bounds in communication networks with variable-rate links," in *Proc. SIGCOMM'95*, Cambridge, MA, 1995, pp. 126–136.

[24] Y. T. Wang and B. Sengupta, "Performance analysis of a feedback congestion control policy under nonnegligible propagation delay," in *Proc. ACM SIGCOMM'91*, 1991, pp. 149–157.

[25] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer," *Adv. Appl. Prob.*, vol. 20, pp. 646–676, 1988.

[26] H. Wang and M. Schwartz, "Comparison of multicast flow control algorithms over combined wired/wireless networks," in *Proc. MoMuc-3*, WINLAB, Princeton, NJ, Sept. 1996, http://www.ctr.columbia.edu/whycu/res.html.

[27] S. Y. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM'96*, San Francisco, CA, Mar. 1996.

[28] M. H. Ammar and L. R. Wu, "Improving the performance of point to multi-point ARQ protocols through destination set splitting," in *Proc. IEEE INFOCOM'92*, Florence, Italy, May 1992, pp. 262–271.

Huayan Wang (S'96) received the B.E. degree from Tsinghua University, Beijing, P.R. China, in 1993 and the M.E. degree from Columbia University, New York, NY, in 1995.

She is currently a Ph.D. candidate in the Department of Electrical Engineering, Columbia University. Her research interests are in the areas of wireless communication networks, with a focus on the research issues in handling the capability mismatch between broad-band and narrow-band wireless networks.

Ms. Wang was a recipient of an Edwin Howard Armstrong Memorial Award for Master Graduates in 1995 from Columbia University. She is a student member of the ACM.

Mischa Schwartz (S'46–A'49–M'54–SM'54–F'66–LF'92) received the B.E.E. degree from Cooper Union, New York, NY, in 1947, the M.E.E. degree from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1949, and the Ph.D. degree in applied physics from Harvard University, Cambridge, MA, in 1951.

From 1947 to 1952, he was a Project Engineer with the Sperry Gyroscope Company, working in the fields of statistical communication theory, radar detection, and radar system design. From 1952 to 1974, he was a Professor of Electrical Engineering at the Polytechnic Institute of Brooklyn, serving as Head of the Department of Electrical Engineering from 1961 to 1965. During the year 1965–1966, he was an NSF Science Faculty Fellow at the Laboratoire de Physique, Ecole Normale Superieure, Paris, France. During the academic year 1973–1974, he was a Visiting Professor at Columbia University, New York, NY, joining that institution in September 1974 as a Professor of Electrical Engineering and Computer Science. He is currently Charles Batchelor Professor of Electrical Engineering. For the 1980 calendar year, he was on leave as a Visiting Scientist with IBM Research. During 1986, he was on leave as a Resident Consultant with NYNEX Science and Technology. During 1994, he spent half-time at IBM Research, working in the field of wireless communication systems. During the first six months of 1995, he was, first, a Visiting Professor at University College, London, under an EPSRC Visiting Fellowship, and then a half-time consultant at AT&T Bell Laboratories. He is author or coauthor of nine books and over 150 technical publications on communication theory and systems, signal processing, and computer communication networks.

Dr. Schwartz is a member of the National Academy of Engineering. He is a former Director of the IEEE, formerly Chairman of its Information Theory Society, and past President of the Communications Society. He is on the Editorial Boards of *Networks*, *Computer Networks and ISDN Systems*, the Japanese journal *IEICE Transactions on Communications*, and the *Journal on Wireless Networks*. He was a recipient of a Distinguished Visitor Award in 1975 from the Australian–American Education Foundation. He is a Fellow of the AAAS, and past Chairman, Commission C, U.S. National Committee/URSI. In 1983, Dr. Schwartz received the IEEE Education Medal. Columbia awarded him its Great Teacher Medal in 1983. In 1984, the IEEE Centennial year, he was cited as one of the ten all-time outstanding electrical engineering educators. He served from 1985–1988 as Director of the Columbia University Center for Telecommunications Research, one of six national Engineering Research Centers established in 1985 under major grants of the National Science Foundation. In 1986, Dr. Schwartz was the recipient of the Cooper Union Gano Dunn Award, given annually for outstanding achievement in science and technology. In 1989, he received the IEEE Region I Award for outstanding engineering management and leadership. In 1994, he received the IEEE Communications Society Edwin H. Armstrong Achievement Award for outstanding contributions to communications technology. In 1995, he received the Mayor's Award for Excellence in Technology, New York City.