Distributed Explicit Rate Schemes in Multi-Input–Multi-Output Network Systems

Naixue Xiong, *Member, IEEE*, Athanasios V. Vasilakos, *Senior Member, IEEE*, Laurence T. Yang, *Member, IEEE*, Yi Pan, *Senior Member, IEEE*, Cheng-Xiang Wang, *Senior Member, IEEE*, and Art Vandenberg, *Member, IEEE*

***Abstract*—With the ever-increasing wireless/wired data applications recently, considerable efforts have focused on the design of distributed explicit rate flow control schemes for multi-input-multi-output service. This paper describes two novel wireless/wired multipoint-to-multipoint multicast flow control schemes, which are based on the distributed self-tuning proportional integrative plus derivative (SPID) controller and distributed self-tuning proportional plus integrative (SPI) controller, respectively. The control parameters can be designed to ensure the stability of the control loop in terms of source rate. The distributed explicit rate SPID and SPI controllers are located at the wireless/wired multipoint-to-multipoint multicast source to regulate the transmission rate. We further analyze the theoretical aspects of the proposed algorithm, and show how the control mechanism can be used to design a controller to support wireless/wired multipoint-to-multipoint multicast transmissions. Simulation results demonstrate the efficiency of the proposed scheme in terms of system stability, fast response, low packet loss, and high scalability, and the results also show SPID scheme has better performance than SPI scheme, however, SPID scheme requires more computing time and CPU resource.**

***Index Terms*—Explicit rate, flow control, multi-input–multi-output (MIMO) system, stability.**

I. Introduction

THE ADVANCES in multi-input–multi-output (MIMO) systems and networking technologies introduced a revolution recently, which promises significant impact in our lives. Especially with ever-increasing multicast data applications, wireless and wired multicast (multipoint-to-multipoint) transmission has considerable effect on many applications such as teleconferencing and information dissemination services. Multicast improves the efficiency of multipoint data distribution from multiple senders to a set of receivers [1]–[5], [21]. Unfortunately, the widely used multicast transport protocols, which are layered on top of IP multicast, can cause congestion or even congestion collapse if adequate flow control is not provided. Flow control thus plays an important role in the traffic management of multicast communications. Without an adequate flow control scheme being implemented in a multicast tree, the incoming traffic to a bottleneck link might be much more than the outgoing link capacity, which could subsequently cause the buffer to overflow, and cause excessive queuing delay or even deadlock in certain nodes.

There are many flow schemes handling unicast transmissions efficiently [6], and they were formulated as a discrete-time feedback control problem with delays. This control-theoretic approach to explicit rate control for available bit rate (ABR) service was further analyzed and verified using a real-network test bed in the work of Kolarov and Ramamurthy [7]. Further achievements in this regard can also be found in [8]–[10]. All these methods are efficient in rate allocation and flow control for unicast transmission. Unfortunately, multicast flow control is much more sophisticated than that of unicast, due to the complexity of multicasting mechanism.

Several multicast flow approaches have been proposed recently. One class of them [11]–[13] adopts a simple hop-by-hop feedback mechanism, in which the feedback, i.e., backward control packets (BCPs), from downstream nodes are initially gathered at branch points, and then are transmitted upward by a single hop upon receipt of a forward control packet (FCP). This kind of manipulation can be carried out on the basis of the tree structure in a multicast transmission. The main merit of these methods lies in the simplicity of the hop-by-hop mechanism. However, at the same time, they often lead to the so-called consolidation noise problem [14], [15] due to incomplete feedback information. To overcome this drawback, Jain *et al.* and Lee *at al*. [15], [16] proposed a method, called feedback synchronization at each branch point, by accumulating feedback from all downstream branches. These schemes then introduce another problem of slow transient response due to the feedback from “long” paths. Such delayed congestion feedback can cause excessive queue buildup/packet loss at bottleneck links.

Manuscript received October 20, 2008; accepted December 2, 2009. Date of publication March 4, 2010; date of current version June 16, 2010. This work was supported by the U.S. National Science Foundation under Grant CNS-0831634. The work of C.-X. Wang was supported by the Scottish Funding Council for the Joint Research Institute in Signal and Image Processing with the University of Edinburgh, which is a part of the Edinburgh Research Partnership in Engineering and Mathematics. This paper was recommended by Associate Editor M.-H. Lim.

N. Xiong is with the Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA (e-mail: nxiongcs@cs.gsu.edu).

A. V. Vasilakos is with the Department of Computer and Telecommunications Engineering, University of Western Macedonia, Kozani 50100, Greece (e-mail: vasilako@ath.forthnet.gr).

L. T. Yang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@stfx.ca).

Y. Pan is with the Department of Computer Science and the Department of Computer Information Systems, Georgia State University, Atlanta, GA 30303 USA (e-mail: pan@cs.gsu.edu).

C.-X. Wang is with the Joint Research Institute for Signal and Image Processing, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, U.K. (e-mail: cheng-xiang.wang@hw.ac.uk).

A. Vandenberg is with the Information Systems and Technology Department, Georgia State University, Atlanta, GA 30302-3965 USA (e-mail: avandenberg@gsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2009.2039872

Xiong *et al.* and Shi and Waldvogel [17], [18] suggested that only a suitable set of representatives, instead of all receivers, sent feedback to the sender, and Xiong *et al.* [19] proposed a fuzzy-logic-based consolidation algorithm to estimate the unknown flow information caused by long propagation delay. More recently, Zhang *et al.* [10] proposed an optimal second-order rate control algorithm to deal with control packet round-trip time (RTT) variations in multicast communications. This method has studied the system dynamics by using the binary congestion feedback in the scenarios of both persistent and on–off elastic traffic services, which defines that the data transfer rate is adjusted at the source, depending on the available bandwidth at the bottleneck. A lot of approaches use queue schemes to solve congestion control problems [25]. Queue schemes in routers make sure that the buffer occupancy stabilizes and never overflows the buffer capacity. Our schemes are based on the explicit rate schemes in the senders. These are active and effective methods to adjust the sending rates, and reduce the packets loss.

The major difficulty in designing multicast flow control protocols arises from the long and heterogeneous RTTs involved in the closed-loop control. In-depth research remains in the following three aspects. First, the existing algorithms usually lack scalability, since they require each router to keep maintaining the saturation status of every session,[1] and as virtual sessions (VSs) travel through it, this yields a major computational bottleneck. To this end, we are going to present an algorithm that is scalable. Second, the known flow control methods usually do not have explicit control over link buffer occupancy; as a consequence, the allocated rate can wander considerably before converging, and the link flow can temporarily exceed the capacity. To attack this problem, we will focus on the stability of our rate control scheme. Third, no explicit rate (ER) allocation has been given in the known approaches. This paper will consider ER-based rate control with an ER feedback (ER value). Such an ER-based scheme is responsive to network congestion and can serve WAN environments quite well where the bandwidth-delay product is usually large.

In this paper, we develop a distributed ER allocation algorithm to overcome the vulnerability due to the heterogeneous multicast receivers. In our scheme, flow controllers regulate the source rate at a multicast tree, which accounts for the buffer occupancies of all destination nodes. The proposed control scheme uses a distributed self-tuning proportional integrative plus derivative (SPID) controller or uses a distributed self-tuning proportional plus integrative (SPI) controller. The control parameters can be designed to ensure the stability of the control loop in terms of source rate. We further show how the control mechanism can be used to design a controller to support multipoint-to-multipoint multicast transmission based on ER feedback. System stability criterion is derived in the presence of destination nodes with heterogeneous RTTs. We analyze the theoretical aspects of the proposed algorithm and verify its agreement with the simulations in the case of a bottleneck link appearing in a multicast tree. Simulation results demonstrate the efficiency of the proposed scheme in terms of system stability and fast response to the buffer occupancy, as well as controlled sending rates, low packet loss, and high scalability. Furthermore, the results also show that SPID scheme has better performance than SPI scheme, though the SPID scheme requires more computing time and CPU resources.

[1]A session is composed of a multicast source, the corresponding transfer path, and the corresponding switch node.

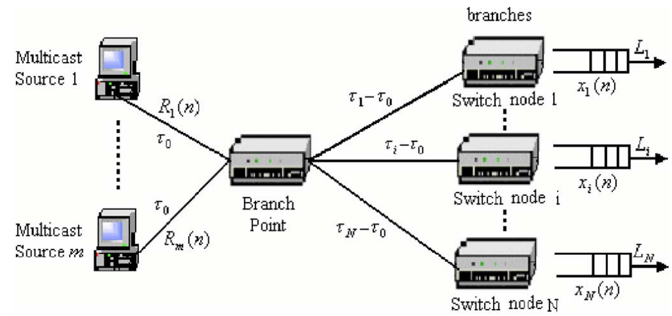


Fig. 1. Multicast configuration of multiple points to multiple points.

The rest of this paper is organized as follows. In Section II, we present the flow control network model. Section III presents two novel schemes, called SPID controller and SPI controller, and describes the specific algorithms of SPID and SPI flow control methods. In Section IV, we analyze the stability of the SPID controller. Section V presents various simulation results to validate and evaluate the performance of our scheme. Finally, we conclude our study and discuss future work in Section VI.

II. Network Configuration Model

To conveniently analyze the performance and characteristics of the proposed multicast scheme, we focus on the following network model [19] (see Fig. 1). The multicast network is a connection-oriented one, which is composed of sources and destination nodes, and time is slotted with the duration $[n, n+1]$ by the sampling period T. The associated data are transferred by a fixed size packet.

In each multicast connection and every sampling period, the multicast source issues and transmits a FCP to the downstream nodes (the branch node and destination nodes), and a BCP is constructed by each downstream node and sent back to the source. After the multicast source receives the BCPs from the downstream nodes, it will take appropriate action to adjust its transmitting rates of multicast traffic based on the computed value of the SPID controller. After receiving the data packets coming from the branch point, the receivers construct BCPs and send them back to the branch point.

The considered multicast service is described as follows. The number of packets sent out by the switch node i in one interval T is denoted by L_i [21], the switch node i has the forward delay τ_i $(1 \le i \le N)$ from sources, and τ_0 is the delay from sources to the neighboring downstream node. Then, the round-trip delay (RTD) for the switch node i is $\tau_{Ri} = 2\tau_i$ and $\tau = \max\{\tau_{R1}, \tau_{R2}, \ldots, \tau_{RN}\}$. We further assume that τ_i and τ_{Ri} are integers, which is reasonable by adjusting T. And the link delay is dominant compared to the other delays, such as proceeding delay, queuing delay, etc. In the model, we assume that $\tau_i \leq \tau_j$ when $1 \leq i \leq j \leq N$. Each branch point schedules the packets in a first-come first-served way. The component $R_i(n)$ represents the receiving rate of the computed receivers i at time slot n.

On the basis of the earlier considerations, the buffer occupancy of the switch node i is determined by [19], [20]

$$x_i(n+1) = \text{Sat}_{K_i}\left\{x_i(n) + \sum_{q_1}^{m} e_q R_q(n-\tau_i) - L_i\right\} \quad (1)$$

where K_i is the buffer size, $x_i(n)$ is the buffer occupancy of the switch node i at time slot n, and $R_q(n-\tau_i)$ is the sending rate of the qth $(1 \leq q \leq m)$ source to the switch node $i (1 \leq i \leq N)$, with

$$\text{Sat}_{K_i}\{x_i\} = \begin{cases} K_i, & x_i > K_i \\ x_i, & 0 \leq x_i \leq K_i \\ 0, & x_i < 0 \end{cases}$$

and

$$e_q = \begin{cases} 1, & \text{if the } q\text{th source is active} \\ 0, & \text{if the } q\text{th source is not active.} \end{cases}$$

After lifting the saturation restriction [6], (1) can be written as

$$x_i(n+1) = x_i(n) + \sum_{q=1}^{m} e_q R_q(n-\tau_i) - L_i. \quad (2)$$

III. SPID and SPI Schemes

The router buffer occupancy is expected to stabilize in the neighborhood of the desired level. If $x(n)$ is too high, it often leads to buffer overflow and packet loss. In addition, under this circumstance, long queuing delay usually results in time out and retransmission, which, in turn, builds up the mounting buffer occupancy; consequently, a vicious cycle is formed. If $x(n)$ is too low, it increases the likelihood of link under utilization during occasionally idle periods. Thus, the router buffer occupancy plays an important role in the congestion control that is chosen to be the feedback carried in BCP.

A. SPID and SPI Algorithms

Generally, among all downstream nodes, the most congested one, defined as the worst node, deserves special attention. On the basis of this consideration, we propose the following SPID control scheme:

$$R_q(n) = \mu + a\sum_{i=1}^{N}(x_i(n-\tau_i) - \bar{x}_i) + \sum_{j=1}^{\tau} b_j R_q(n-j) + c\sum_{k=1}^{N}(x_k(n-\tau_k) - x_k(n-\tau_k-1)) \quad (3)$$

where $a, b_j \ (j = 1, 2, \ldots, \tau)$, and c are the proportional, integral, and derivative control gains, respectively, which are to be determined by the stability criteria. These coefficients are used to locate all the poles of the closed-loop equations (2) and (3) within the unit circle to ensure stability. The component $\bar{x}_i$ is the target queue length and μ is the maximum sending rate of sources.

To save computing time and CPU resources, here we present a simple SPI control scheme as follows:

$$R_q(n) = \mu + a\sum_{i=1}^{N}(x_i(n-\tau_i) - \bar{x}_i) + \sum_{j=1}^{\tau} b_j R_q(n-j) \quad (4)$$

where a and $b_j \ (j = 1, 2, \ldots, \tau)$ are the proportional and integral control gains, respectively, which are to be determined from the stability criteria based on control theory. Similarly, these coefficients should make all the poles of the closed-loop equations (2) and (4) within the unit circle to ensure stability.

In (3) and (4), if the buffer occupancy of the switch node i is measured at the instances $n - \tau_i$, after the feedback delay τ_i, the BCP reaches the controller located at the source $q \ (q = 1, 2, \ldots, m)$, and the source then takes out the buffer occupancy of the destination nodes at time slot $t = n$. By doing so, the proposed controller can be expected to have flexibility to cope with the sharp oscillation in buffer occupancy that could cause the network to lose packets. In addition, the calculation in (3) and (4) is completely independent of virtual connections traveling through the multicast session. This means the scheme has scalability.

B. Implementation of the SPID and SPI Algorithms

Each branch point of the multicast tree replicates each data packet and FCP from its upstream node to all its downstream branches. The downstream nodes return their congestion information via BCPs to the parents through the backward direction once they receive FCPs. Moreover, the branch nodes consolidate the BCPs that carry all the available rates and the relevant link bandwidth from different branches into one BCP and feedback the new BCP to their upstream node. Associated rate adaptors are located at multicast sources.

There is a single first-in first-out (FIFO) queue to multiplex all flows traveling through the outgoing link. Assume that congestion never happens at the router connected with the sources; hence, these two can be consolidated into one node, which is true in most cases in real networks.

Before we present the algorithm in detail, we specify the following variables. The variable multicasttree$[i] = 1(0)$ means the ith branch point receives (does not receive) FCP or BCP control packet, while receivertree$[j] = 1(0)$ means the jth branch point receives (does not receive) confirmations from all destination nodes. On the basis of the earlier specifications, the pseudocode of the proposed router and source algorithms in congestion control model of multiple points to multiple points in a multicast network is shown in Fig. 2.

IV. System Stability Analysis

The ability of a multicast tree to provide efficient heterogeneous distributed communication that can guarantee multiple

Source Algorithm

- Upon every T sampling period
  - Transmit data streams including FCP;
- Upon receipt of a BCP from its downstream
  - Compute the sending rate based on the BCP using SPID/SPI controller;
  - Adjust the transmitting rates based on computed sending rate.

Router Algorithm

Variables:
$Multicasttree[i] = 1(0)$;
// the i^{th} router receives (doesn't receive) FCP or BCP control packet;
$Receivertree[j] = 1(0)$;
// the j^{th} router receives (doesn't receive) confirmations of all receivers;
If $Multicasttree[i] = 1$

- If the packet is an FCP
  - Put the data packet in the buffer;
  - Copy the data including FCP;
  - Multicast them to the downstream nodes;
- If the packet is a BCP
  - Construct the BCP based on the received BCPs;
  - Feed it back to the upstream node;
  - If $Receivertree[j] = 1$
    - Delete the data packets from the buffer;
  - If $Receivertree[j] = 0$
    - Maintain the data packets in the buffer until all confirmations of the receivers are received;

Receiver Algorithm

- Upon receipt of an FCP
  - Put the data packets into the buffer;
  - Construct the BCP based on the current flow case of the receiver;
  - Feedback the BCP to the upstream routers.

Fig. 2. Pseudocode of the source/router/destination node algorithms.

qualities of service can only be realized by effective traffic management schemes.

In this paper, a rate-based scheme is used rather than a window-based adaptation algorithm to achieve congestion control in multirate–multicast control (MR-MCC) tree. The window-based scheme has extra complexity in maintaining and synchronizing the congestion window across all receivers, and it usually generates data bursts periodically [22]. In our proposed SPID and SPI control schemes, the rate adaptation takes into account the buffer occupancies of the destination nodes as well as the variation of RTTs. The controller parameters are designed to guarantee the stability of rate, which ensures a smooth dynamic rate adaptation to minimize packet loss rate. This, in turn, brings an obvious advantage of the proposed scheme over the widely adopted additive increase and multiplicative decrease (AIMD) (see, for example, [22]). For example, in AIMD, it is difficult to choose the appropriate increase and decrease factors to guarantee the system's stability, and then to obtain smooth and healthy rate adaptation and good link utilization.

A. SPID System Stability Analysis

In this section, the stability of the proposed SPID congestion control scheme is analyzed as follows. Considering (2), if z transformation is applied, one can easily arrive at

$$(z-1)X_i(z) = \sum_{q=1}^{m} e_q R_q(z) z^{-\tau_i} - L_i D(z) \tag{5}$$

where the z transformation of $x_i(n)$ and $R_j(n)$ are, respectively, described by $X_i(z) = \sum_{n=0}^{+\infty} x_i(n) z^{-n}$, $R_q(z) = \sum_{n=0}^{+\infty} R_q(n) z^{-n}$, and $D(z) = \sum_{n=0}^{+\infty} z^{-n} = z/z-1$.

Taking the z transform of (3), it yields

$$R_q(z) = \mu D(z) + a\sum_{i=1}^{N}\left[z^{-\tau_i} X_i(z) - \bar{x}_i D(z)\right] + \sum_{j=1}^{\tau} b_j z^{-j} R_q(z) + c\sum_{k=1}^{N}\left(z^{-\tau_i} X_i(z) - z^{-\tau_i - 1} X_i(z)\right). \tag{6}$$

From (5) and (6), one has

$$\Delta_1(z) R_q(z) = a\sum_{i=1}^{N}\left(-L_i D(z) z^{-\tau_{Ri}/2} - \bar{x}_i D(z)(z-1)\right) - c\sum_{k=1}^{N} z^{-\tau_{Ri/2}} L_i D(z)(1-z^{-1}) + \mu D(z)(z-1) \tag{7}$$

where we denote

$$\Delta_1(z) = \left(1 - \sum_{j=1}^{\tau} b_j z^{-j}\right)(z-1) - a\sum_{i=1}^{N} z^{-\tau_{Ri}} - c\sum_{k=1}^{N}\left(z^{-\tau_{Rk}} - z^{-\tau_{Rk}-1}\right). \tag{8}$$

The coefficients a, b_j $(j = 1, 2, \ldots, \tau)$, and c are determined by the stability criteria of the control theory.

The component $\Delta_1(z)$ is the *characteristic polynomial* (CP) of the multicast system given by (2) and (3) [23]. The CP (8) is closely related to the stability of the congestion-controlled network system. From a control-theoretic view, when all the zeros of (8) lie within the unit disc, the original network system (2) with the controller (3) is stable in terms of the source sending rates. Stability is a prerequisite in congestion control to ensure that the network has no oscillation of sending rate, and thus minimizes the packet loss rate.

The following computation procedures aid us in determining if all the roots of a CP lie within a unit disc.

For our purpose, we use Schur–Cohn stability test here. Generally, for a polynomial

$$A_Q(z) = 1 + \sum_{n=1}^{Q} a_n^{(Q)} z^{-n} \tag{9}$$

we need to know if all its zeros lie in the unit disc. The following algorithm is proposed for this purpose.

Algorithm 1: Schur–Cohn stability test

Step 1) Start with the original polynomial of degree Q (Q coefficients).

Step 2) Generate a sequence of polynomials recursively. $A_i(z)$, $i = Q : -1 : 0$, according to

$$A_{i-1}(z) = (A_i(z) - q_i z^{-i} A_i(z^{-1}))/(1 - q_i^2) \quad (10)$$

where $q_i = a_i^{(i)}$. Note that $z^{-i}A_i(z^{-1})$ is a flipped version of $A_i(z)$.

Step 3) The zeros of the polynomial $A_Q(z)$ are inside the unit circle iff

$$|q_i| < 1, \quad i = Q : -1 : 1. \quad (11)$$

The previous Algorithm 1 is devised to implement the aforementioned stability condition.

With regard to CP (8), some manipulations are needed for it to satisfy the form of the polynomial $A_Q(z)$. This is done in the following manner:

$$\Delta_1(z) = z[1 - (1 + b_1)z^{-1} + (b_1 - b_2)z^{-2} + \cdots + (b_{\tau-1} - b_\tau)z^{-\tau} + (b_\tau - a - c)z^{-\tau-1} + cz^{-\tau-2}]$$

where we denote

$$\Delta_1'(z) = 1 - (1 + b_1)z^{-1} + (b_1 - b_2)z^{-2} + \cdots + (b_{\tau-1} - b_\tau)z^{-\tau} + (b_\tau - a - c)z^{-\tau-1} + cz^{-\tau-2}. \quad (12)$$

Therefore, we have

$$\Delta_1(z) = z\Delta_1'(z).$$

It is sufficient to ensure the stability if all the roots of $\Delta_1'(z)$ lie inside the unit circle because $\Delta_1(z)$ has all roots of $\Delta_1'(z)$ together with a root $z = 0$, which is obviously inside the unit circle. Thus, Schur–Cohn criterion can be applied to $\Delta_1'(z)$ directly with $Q = \tau + 2$.

Without loss of generality, we group these nodes into one class, with a small variation of time delays and sending rates. Thus, we divide N destination nodes into M groups based on the RTTs, and in each group, the RTT is assumed to be equal, i.e.,

$$\{\tau_{R1}, \tau_{R2}, \tau_{R3}, \ldots, \tau_{RN}\} = \{\tau_1, \ldots \tau_1, \tau_2, \ldots \tau_2, \ldots, \tau_M, \ldots, \tau_M\} \quad (13)$$

and we set n_i as the number of the RTT t_i ($i = 1, 2, \ldots, M$) corresponding to the ith group receivers, then $N = \sum_{i=1}^{q} n_i$, where n_i is a positive integer. So, the CP (12) is

$$\begin{aligned}\Delta_1'(z) = z[&1 - (b_1 + 1)z^{-1} + \cdots + (b_{\tau_{R1-1}} - b_{\tau_{R1}}) \\ &+ (b_{\tau_{R1}} - b_{\tau_{(R1)+1}} - an_1 - cn_1)z^{-(\tau_{R1}-1)} \\ &+ (b_{(\tau_{R1+1})} - b_{(\tau_{R1})+2} + cn_1)z^{-(\tau_{R1}-2)} \\ &+ (b_{(\tau_{Rn_i}-1)} - b_{\tau_{Rn_i}})z^{-\tau_{Rn_i}} \ldots \\ &+ (b_{\tau_{Rn_i}} - b_{\tau_{(Rn_i)}+1} - an_i - cn_i)z^{-\tau_{Rn_i}-1} \\ &+ (b_{\tau_{(Rn_i)}+1} - b_{\tau_{(Rn_i)}+2} + cn_i)z^{-\tau_{(Rn_i)}-2} + \cdots \\ &+ (b_{\tau-1} - b_\tau)z^{-\tau} + (b_\tau - an_M - cn_m)z^{-\tau-1} \\ &+ cn_M z^{-\tau-2}]. \end{aligned} \quad (14)$$

Let

$$\begin{cases} b_1 + 1 = \varepsilon \\ b_2 - b_1 = \varepsilon \\ b_3 - b_2 = \varepsilon \\ \ldots \\ b_{\tau_{R1}} - b_{\tau_{R1}-1} = \varepsilon \\ an_1 + b_{\tau_{R1}+1} + cn_1 - b_{\tau_{R1}} = \varepsilon \\ b_{\tau_{R1}+2} - cn_1 - b_{\tau_{R1}+1} = \varepsilon \\ \ldots \\ b_{\tau_{Rn_i}} - b_{(\tau_{Rn_i}-1)} = \varepsilon \\ b_{(\tau_{Rn_i}+1)} - b_{\tau_{Rn_i}} + an_i + cn_i = \varepsilon \\ b_{\tau_{Rn_i}+2} - b_{\tau_{Rn_i}+1} - cn_i = \varepsilon \\ \ldots \\ b_\tau - b_{\tau-1} = \varepsilon \end{cases} \quad (15)$$

where $i = 1, 2, 3, \ldots, (M - 1)$, and

$$an_M + cn_M - b_\tau = \varepsilon \quad (16)$$

$$-cn_M = \varepsilon. \quad (17)$$

Then we can obtain

$$a = (\tau\varepsilon + 2\varepsilon - 1)/N \quad (18)$$

$$c = (-\varepsilon)/n_M \quad (19)$$

and

$$b_j = \begin{cases} j\varepsilon - 1;\ (j = 1, 2, 3, \ldots, \tau_{R1}) \\ j\varepsilon - 1 - a(n_1 + n_2 + \cdots + n_i) - cn_i; \\ \qquad (j = \tau_{(Rn_i)} + 1) \\ j\varepsilon - 1 - a(n_1 + n_2 + \cdots + n_i);\ (j = \tau_{Rn_i} + 2) \\ j\varepsilon - 1 - a(n_1 + n_2 + \cdots + n_i);\ (j \in (\tau_{R1} + 1, \tau]; \\ \qquad \text{and } j = \tau_{Rn_i} + 3, \ldots, \tau_{Rn_{(i+1)}}) \end{cases} \quad (20)$$

where $i = 1, 2, \ldots, (M - 1)$. We can set

$$\Delta_1'(z) = z^{-\tau_{Rj}}[z^{(\tau_{Rj}+1)}, \ldots, -\varepsilon(z^{\tau_{Rj}} + z^{(\tau_{Rj}-1)} + z^{(\tau_{Rj}-2)} + \cdots + z + 1)]. \quad (21)$$

From [24], when $\epsilon < 1/(\tau + 2)$, all the zeros of (21) lie within the unit disc, and the original network system (2) with the controller (3) is stable.

B. SPI System Stability Analysis

On the basis of the aforementioned SPID stability analysis, we use a similar method to analyze SPI system stability.

In order to analyze the stability, first we consider (4); if z transformation is applied, one easily arrives at

$$R_q(z) = \mu D(z) + a\sum_{i=1}^{N}\left[z^{-\tau_i}X_i(z) - \bar{x}_i D(z)\right] + \sum_{j=1}^{\tau} b_j z^{-j} R_q(z). \tag{22}$$

From (5) and (22), one has

$$\Delta_2(z)R_q(z) = a\sum_{i=1}^{N}\left(-L_i D(z) z^{-\tau_{Ri}/2} - \bar{x}_i D(z)(z-1)\right) + \mu D(z)(z-1) \tag{23}$$

where we denote

$$\Delta_2(z) = \left(1 - \sum_{j=1}^{\tau} b_j z^{-j}\right)(z-1) - a\sum_{i=1}^{N} z^{-\tau_{Ri}}. \tag{24}$$

The coefficients a and b_j $(j = 1, 2, \ldots, \tau)$ are determined by the stability criteria of the control theory. The component $\Delta_2(z)$ is the CP of the multicast system given by (2) and (4) for SPI control system [23].

Based on (13), the CP (24) is

$$\begin{aligned}\Delta_2(z) = -z^{-\tau}[&-z^{\tau+1} + (b_1+1)z^{\tau} + (b_2 - b_1)z^{\tau-1} \\ &+ (b_3 - b_2)z^{\tau-2} + \cdots + (b_{\tau_{R1}} - b_{(\tau_{R1})-1})z^{\tau-(\tau_{R1})+1} \\ &+ (b_{(\tau_{R1})+1} - b_{\tau_{R1}} - an_1)z^{(\tau-\tau_{R1})} \\ &+ (b_{(\tau_{R1}+2)} - b_{(\tau_{R1})+1})z^{\tau-(\tau_{R1})-1} + \cdots \\ &+ (b_{\tau_{R_i}} - b_{(\tau_{R_i})-1})z^{\tau-(\tau_{R_i})+1} \\ &+ (b_{\tau(R_i+1)} - b_{(\tau_{R_i})} - an_i)z^{\tau-(\tau_{R_i})} + \cdots \\ &+ (b_{\tau_R(M-1)} - b_{(\tau_R(M-1))-1} - an_{(M-1)})z^{\tau-(\tau_R(M-1))} \\ &+ \cdots + (b_{\tau} - b_{\tau-1})z + (b_{\tau} - an_M].\end{aligned} \tag{25}$$

Let

$$\begin{cases} b_1 + 1 = \varepsilon \\ b_2 - b_1 = \varepsilon \\ b_3 - b_2 = \varepsilon \\ \cdots \\ b_{\tau_1} - b_{(\tau_1)-1} = \varepsilon \\ b_{(\tau_1)+1} - b_{(\tau_1)} - an_1 = \varepsilon \\ b_{(\tau_1)+2} - b_{(\tau_1)+1} = \varepsilon \\ \cdots \\ b_{(\tau_{R_i})} - b_{(\tau_{R_i})-1} = \varepsilon \\ b_{(\tau_{R_i})+1} - b_{(\tau_{R_i})} - an_i = \varepsilon \\ \cdots \\ b_{(\tau_{R_{(M-1)}})} - b_{(\tau_{R_{(M-1)}})-1} - an_{(M-1)} = \varepsilon \\ \cdots \\ b_{\tau} - b_{(\tau-1)} = \varepsilon \end{cases} \tag{26}$$

and

$$b_{\tau} = an_M. \tag{27}$$

We obtain

$$a = (\tau\varepsilon - 1)/N \tag{28}$$

and

$$b_j = \begin{cases} j\varepsilon - 1, \quad (j = 1, 2, 3, \ldots, \tau_{R1}) \\ j\varepsilon - 1 - a(n_1 + n_2 + \cdots + n_i), \quad i \in [1, (M-1)], \\ \qquad j \in [\tau_{(R_i)+1}, \tau_{R_{(i+1)}}]). \end{cases} \tag{29}$$

We can get

$$\Delta_2(z) = -z^{-\tau}\left[-z^{(\tau+1)} + \varepsilon(z^{\tau} + z^{(\tau-1)} + z^{(\tau-2)} + \cdots + z)\right]. \tag{30}$$

From [24], when $\epsilon < 1/(\tau+1)$, all the zeros of (30) lie within the unit disc, and the original network system (2) with the controller (4) is stable.

V. Performance Evaluation

To evaluate the performance of the studied multicast congestion control scheme, we focus on the following two simulation models, and are mostly interested in analyzing the transient behaviors of the network. In the performance analysis, the duration of response time, receiving rate of receivers, and steady state of buffer occupancy are the main concerns.

From the view of control theory, a control scheme with short response time has the following advantages: when the buffer of receiver nodes is close to the threshold, one may tell the sending node to reduce the sending rate and prevent the loss of packets as soon as possible; while when the available bandwidth increases, the sending node increases the sending rate as soon as possible and enhances the utilization rate of the bandwidth.

In simulations, we process the nodes that have small differences of time delay and sending rate together. Then we unify the time delay and sending rate. Since the situation of every node in each group (about 20 receivers) is similar, we only choose one node from each group as a representative. We assume that the link delay is dominant compared to the other delays, such as processing delays and queuing delay.

Simulations are carried out over a wide range of patterns, and propagation between two different nodes can lie in the LAN case or the WAN case. Based on the different models and network dynamic behavior, we present two simulation experiments. In simulation 1 (see Fig. 3), the multicast source S1 sends data packets at 0 ms and the multicast source S2 starts to send data packets at 1000 ms in the simulation time; then the joining of S2 enhances the network dynamic behavior, and also demonstrates the efficiency of the SPID and SPI schemes. In simulation 2 (see Fig. 9), there are more receivers and longer delay than in model 1, and we set appropriate parameters to enable system stability. In each model, we compare and evaluate the performance of the SPID and SPI systems.

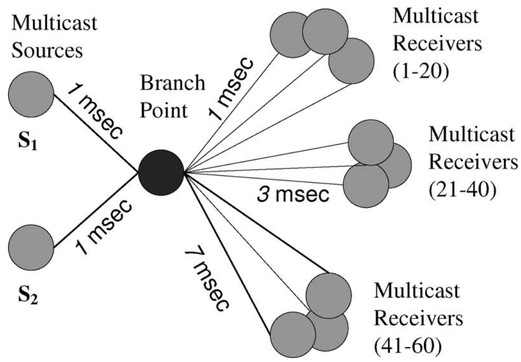


Fig. 3. Multicast simulation model 1.

TABLE I
PARAMETERS IN THE SIMULATION MODEL 1

Parameters	Receiver 1	Receiver 21	Receiver 41
$\bar{x}_i$ (Mb)	70	80	120
L_i (Mbps)	2	3	4
τ_{Rji} (msec)	4	8	16

A. Simulation 1

In this simulation, we focus on comparing the transient behavior of the SPID and SPI network systems based on Fig. 3.

The relevant notations and assumptions are listed in Table I, and $\tau = 16$ ms, $N = 60$, and $n_1 = n_2 = n_3 = 20$. According to the aforementioned simulation parameters and system stability analysis in Section IV to select the control gains, one computes the relevant control parameters based on Functions 18–20 (for SPID scheme) and Functions 28–29 (for SPI scheme).

For SPID scheme, we set ε to be $1/20$, which is stable in the system. Then, $a = -1/600$, $c = -1/400$, and $b = [b_1, b_2, b_3, \ldots, b_{16}]$, i.e.,

$$b = [-19/20, -9/10, -17/20, -8/10, -2/3, -2/3, -37/60, \\ -34/60, -26/60, -26/60, -23/60, -20/60, -17/60, \\ -14/60, -11/60, -7/60, -8/60].$$

For SPI scheme, we also set ε to be $1/19$, which is stable in the system. Then, $a = -1/380$ and $b = [b_1, b_2, b_3, \ldots, b_{16}]$, i.e.,

$$b = [-18/19, -17/19, -16/19, -15/19, -13/19, -12/19, \\ -11/19, -10/19, -9/19, -8/19, -7/19, -6/19, \\ -4/19, -3/19, -2/19, -1/19].$$

In this section, the simulation results of simulation 1 are shown in Figs. 4–8. The sending rates of sources S_1 and S_2 are shown in Figs. 4–5, respectively. The initial sending rate of multicast source S_1 is 6 Mb/s. It can be seen that, although the sending rate of the multicast source S_1 has some fluctuation at first, as time goes on, the sending rate is gradually adjusted and quickly at the value of 2 Mb/s within 74.1 ms for SPID controller, and 325.4 ms for SPI controller. When the multicast source S_2 starts to send data packets at 1000 ms, the sending rate of source S_1 has some fluctuation in response to the multicast source S_2 joining, and quickly stabilizes at the new value of 1 Mb/s within 287.5 ms for SPID controller and 362.5 ms for SPI controller. Figs. 6–8 show the buffer transient responses of the receiver node 1, node 21, and node 41, respectively. In Fig. 6, these buffer occupancies of the bottleneck receivers all have some fluctuation in the beginning. Then they gradually become stable at the value 27.5 Mb for SPID controller and 230 Mb for SPI controller. When the multicast source S_2 starts

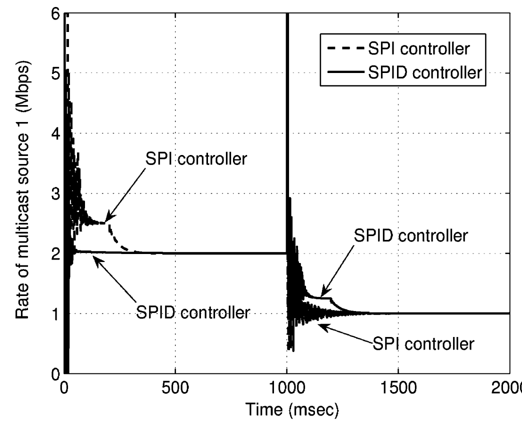


Fig. 4. Sending rate of S1.

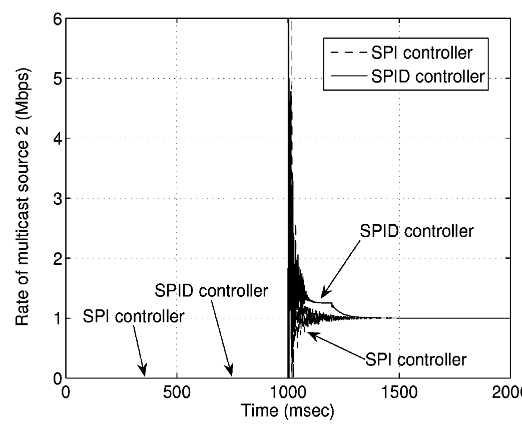


Fig. 5. Sending rate of S2.

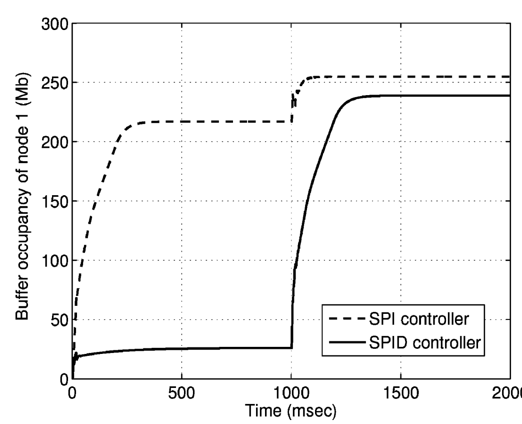


Fig. 6. Buffer occupancy of receiver 1.

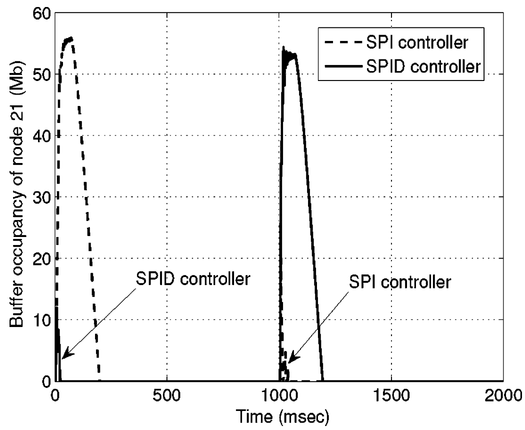


Fig. 7. Occupancy of receiver 21.

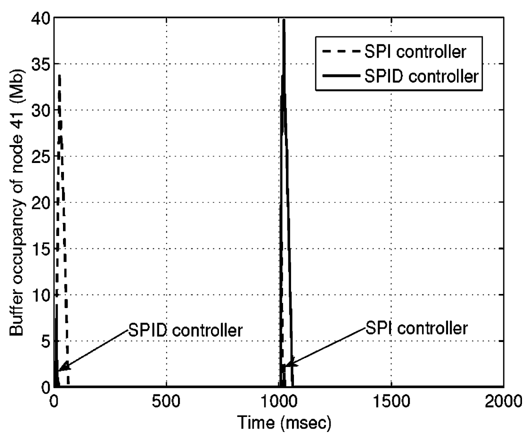


Fig. 8. Occupancy of receiver 41.

TABLE II
COMPARATIVE ANALYSIS OF THE SIMULATION MODEL 1

Parameters	SPID: S_1	SPI: S_1	SPID: S_2	SPI: S_2
Stable sending rate 1: [0, 1000] (Mbps)	2	2	0	0
Response of stable sending rate 1 (msec)	74.1	325.4	0	0
Stable sending rate 2: [1000, 2000] (Mbps)	1	1	1	1
Response of stable sending rate 2 (msec)	287.5	362.5	312.5	325

to send data packets at 1000 ms, the buffer occupancies of the receivers have some fluctuation in response to the multicast source S_2 joining. For receiver group 1, they quickly become stable at the value 240 Mb for SPID controller and 255 Mb for SPI controller. In Figs. 7–8, some packets accumulate in the buffer of receiver 21 and receiver 41 at the beginning. As time goes on, the controller starts to adjust the transmission rate of the source, and the remaining packets in the buffer are cleared.

Table II gives the comparative analysis of the simulation model 1. The second and fourth rows are about the stable sending rates of sources S_1 and S_2 in time range [0, 2000] ms. The third and fifth rows are about the response times of stable sending rates in the two different time ranges. It is clear that our two schemes could quickly adjust the buffer occupancy and the rates of sending node based on the dynamic network environment.

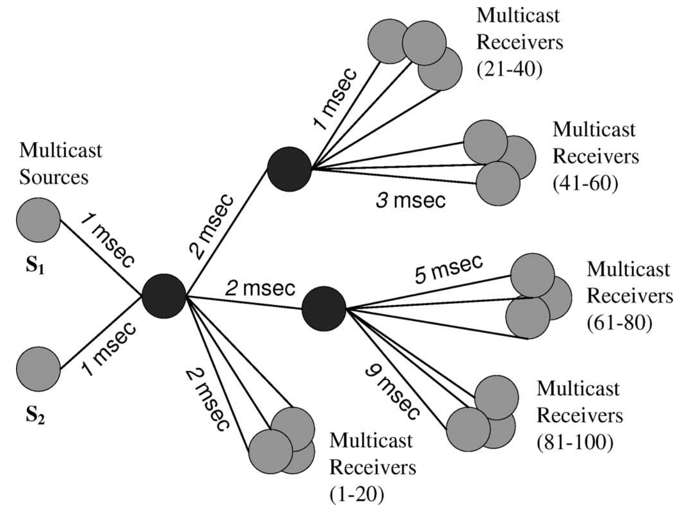


Fig. 9. Multicast system model 2.

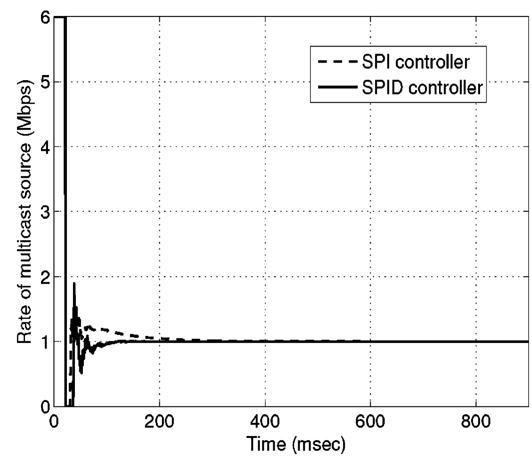


Fig. 10. Sending rate of S1.

These simulation results demonstrate our SPID and SPI controllers, efficiency in terms of system stability, fast response, low packet loss, and high scalability. Based on our schemes, the source adjusts the sending rate gradually to stabilize the buffer occupancy and rate of sending node quickly. Based on the earlier comparative analysis in Table II, we find that SPID controllers can provide better performance than SPI controllers in terms of the fast response of the controlled sending rates and low buffer occupancy of bottleneck receivers.

B. Simulation 2

In this simulation, we focus on comparing the SPID and SPI performances based on the multicast system shown in Fig. 10.

The relevant network parameters and assumptions are listed in Table III, and $\tau = 24$ ms, $N = 100$, and $n_1 = n_2 = \cdots = n_5 = 20$. According to the earlier simulation parameters and system stability analysis in Section IV to select the control gains, the relevant control parameters are computed based on

TABLE III
PARAMETERS IN THE SIMULATION MODEL 2

Parameters	Rece. 1	Rece. 21	Rece. 41	Rece. 61	Rece. 81
$\bar{x}_i$ (Mb)	70	75	80	120	140
L_i (Mbps)	2	2	3	4	5
τ_{Rji} (msec)	6	8	12	16	24

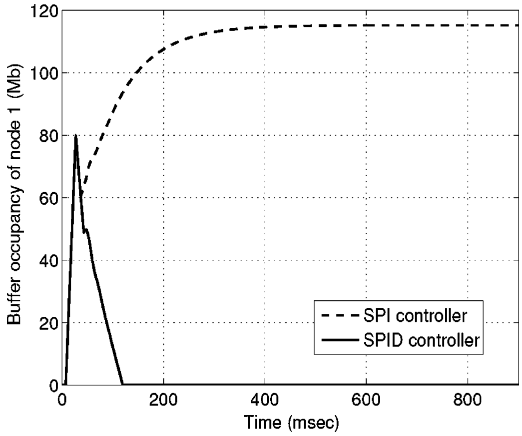


Fig. 11. Occupancy of receiver 1.

Functions 18–20 (for SPID scheme) and Functions 28–29 (for SPI scheme).

For SPID scheme, we set ε to be $1/29$, which is stable in the system. Then, $a = -3/2900$, $c = -1/580$, and $b = [b_1, b_2, \ldots, b_{24}]$, i.e.,

$$b = [-28/29, -27/29, -26/29, -25/29, -24/29, -23/29, -102/145, -102/145, -178/290, -178/290, -168/290, -158/290, -132/290, -132/290, -122/290, -112/290, -102/290, -92/290, -82/290, -72/290, -62/290, -52/290, -42/290, -32/290].$$

For SPI scheme, we set ε to be $1/30$, which is stable in the system. Then, $a = -1/500$ and $b = [b_1, b_2, \ldots, b_{24}]$, i.e.,

$$b = [-29/30, -14/15, -9/10, -13/15, -5/6, -4/5, -109/150, -52/75, -31/50, -44/75, -83/150, -39/75, -67/150, -31/75, -19/50, -26/75, -41/150, -6/25, -31/150, -13/75, -7/50, -8/75, -11/150, -1/25].$$

In this section, as the two sources send data packets at the same time, we only choose the source S_1 as a representative.

The sending rate of multicast source S_1 is shown in Fig. 10. After 74.3 ms for SPID controller and 222.1 ms for SPI controller, the sending rate of source S_1 in stable system is stable at the value 1 Mb/s. Figs. 11–15 show the cases of buffer occupancies in receiver 1, receiver 21, receiver 41, receiver 61, and receiver 81. In stable case, the buffer occupancies in receiver 1 and receiver 21 quickly stabilize at the values 115.5 and 115.1 Mb for SPI controller shown in Figs. 11–12, while SPID controller clears the data in the buffer during the stable period. In Figs. 13–15, some packets accumulate in the buffers of receivers in the beginning. As time goes on, the controller starts to adjust the transmission rate of the source, and the remaining packets in the buffer are cleared.

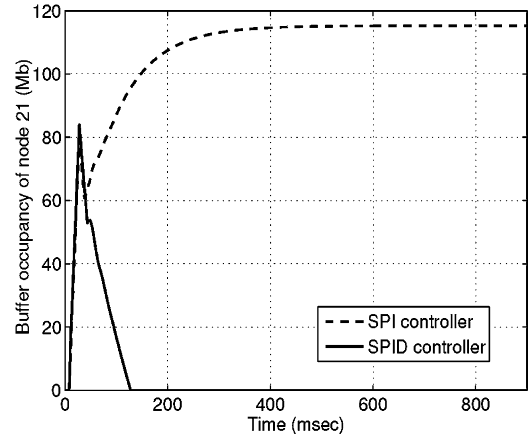


Fig. 12. Occupancy of receiver 21.

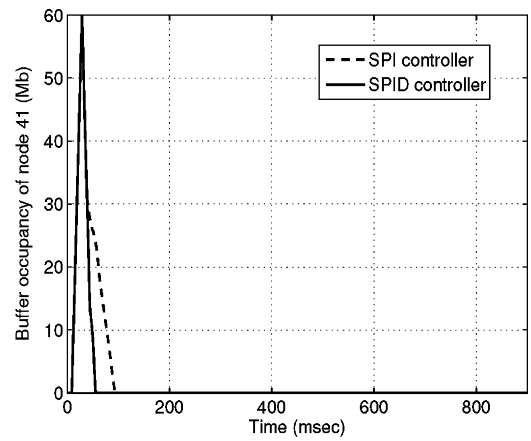


Fig. 13. Occupancy of receiver 41.

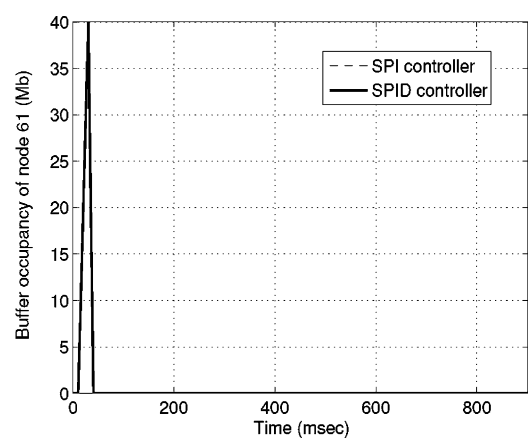


Fig. 14. Occupancy of receiver 61.

These simulation results demonstrate our SPID and SPI controllers, efficiency in terms of system stability, fast response,

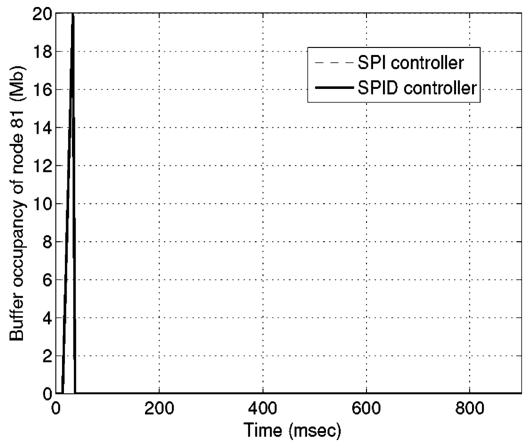


Fig. 15. Occupancy of receiver 81.

TABLE IV
COMPARATIVE ANALYSIS OF THE SIMULATION MODEL 2

Parameters	SPID: S_1	SPI: S_1	SPID: S_2	SPI: S_2
Sending rate in stable period (Mbps)	1	1	1	1
Response time of stable sending rate (msec)	74.3	222.1	74.3	222.1

low packet loss, and high scalability. Based on our schemes, the source adjusts the sending rate gradually to stabilize the buffer occupancy and rate of sending node quickly. Then the theoretical agreement with the simulations in the case of bottleneck link appearing in a multicast tree is verified. Therefore, the SPID and SPI schemes are effective and efficient schemes. Based on the comparative analysis presented in Table IV, we find that SPID controllers can provide better performance than SPI controllers in terms of the fast response of the controlled sending rates and low buffer occupancy of bottleneck receivers.

In Tables II and IV, it is clear that SPID can make faster response in adjusting the sending rates of multicast sources not only at the beginning of the multicast system, but also during the dynamic change of the frame of the multicast system. Thus, based on the earlier comparative analysis, we demonstrate that SPID controller can obtain better multicast performance than SPI controller; however, the SPID scheme requires more computing time and CPU resources.

VI. CONCLUSION

The advances in MIMO systems and networking technologies introduced a revolution in recent times, which promises significant impact throughout our lives, especially in wireless and wired multicast (multipoint-to-multipoint) transmission field.

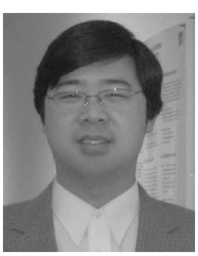
In this paper, we presented two novel wireless and wired multicast schemes, called SPID and SPI schemes, using an explicit rate feedback mechanism to design a controller for regulating the source rates in wireless and wired multipoint-to-multipoint multicast networks. The control parameters of the SPID and SPI controllers can be designed to ensure the stability of the control loop in terms of buffer occupancy and adjust automatically, depending on the network load. This subsequently means that the schemes provide the least packet loss in steady state. Relevant pseudocodes for implementation have been developed, and the paper shows how the two controllers could be designed to adjust the rates of data service. Simulations have been carried out with wireless and wired multipoint-to-multipoint multicast models to evaluate the performance of the SPID and SPI controllers. The simulation results clearly demonstrate the efficiency of our scheme in terms of system stability and fast response of the buffer occupancy, as well as controlled sending rates, low packet loss, and high scalability. The simulation results also show that SPID scheme has better performance than SPI scheme; however, SPID scheme requires more computing time and CPU resources. As evident from the analyses and simulation results, the proposed multicast scheme is simple and also can support unicast. We believe that our study is a valuable foundation for a unified flow control scheme capable of being deployed in real multicast networks.

A limitation of the explicit rate schemes is that if the network has a larger transfer delay, then the effect of the control schemes becomes weak. A possible reason is that a larger delay makes the response time too long, which is not good for an applicable network. Our further research along this line of study would investigate TCP-friendly related issues in multicast congestion control.

REFERENCES

[1] S. Deering, "Host extensions for IP multicasting," Stanford Univ., Stanford, CA, Tech. Memo. RFC1112, Jan. 1989.
[2] R. H. Gau, Z. J. Haas, and B. Krishnamachari, "On multicast flow control for heterogeneous receivers," *IEEE/ACM Trans. Netw.*, vol. 10, no. 1, pp. 86–101, Feb. 2002.
[3] L. Roberts, "Rate based algorithm for point to multipoint ABR service," in *ATM Forum Contribution 94–0772*, Sep. 1994.
[4] K. Y. Siu and H. Y. Tzeng, "On max-min fair congestion control for multicast ABR services in ATM," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 3, pp. 545–556, Apr. 1997.
[5] H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe, "Performance issues in public ABR service," *IEEE Commun. Mag.*, vol. 34, no. 11, pp. 40–48, Nov. 1996.
[6] X. Zhang and K. G. Shin, "Statistical analysis of feedback synchronization signaling delay for multicast flow control," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1152–1161.
[7] Y. Z. Cho, S. M. Lee, and M. Y. Lee, "An efficient rate-based algorithm for point-to-multipoint ABR service," in *Proc. IEEE GLOBECOM*, Nov. 1997, pp. 790–795.
[8] W. Ren, K. Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," presented at the IEEE ATM Workshop, San Francisco, CA, Aug. 1996.
[9] D. DeLucia and K. Obraczka, "Multicast feedback suppression using representatives," in *Proc. IEEE INFOCOM*, 1997, vol. 2, pp. 463–470.
[10] X. Zhang, K. G. Shin, D. Saha, and D. D. Kandlur, "Scalable flow control for multicast ABR services in ATM networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 1, pp. 67–85, Feb. 2002.
[11] L. Benmohamed and S. M. Meekov, "Feedback control of congestion in packet switching networks: The case of a single congested node," *IEEE/ACM Trans. Netw.*, vol. 1, no. 6, pp. 693–708, Dec. 1993.
[12] A. Kolarov and G. Ramamurthy, "A control theoretic approach to the design of an explicit rate controller for ABR service," *IEEE/ACM Trans. Netw.*, vol. 7, no. 5, pp. 741–753, Oct. 1999.
[13] N. X. Xiong, Y. He, L. T. Yang, and Y. Yang, "A self-tuning reliable dynamic scheme for multicast flow control," in *Proc. 3rd Int. Conf. Ubiquitous Intell. Comput. (UIC 2006)*, Wuhan, China, Sep. 3–6, pp. 351–360.
[14] F. Blanchini, R. L. Cigno, and R. Tempo, "Robust rate control for integrated services packet networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 5, pp. 644–652, Oct. 2002.

[15] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA switch algorithm: A complete description," in *ATM Forum-TM*, Aug. 1996, pp. 96–1172.
[16] S. H. Lee and J. T. Lim, "Multicast ABR service in ATM networks using a fuzzy-logic-based consolidation algorithm," *Proc. Inst. Electr. Eng. Commun.*, vol. 148, no. 1, pp. 8–13, Feb. 2001.
[17] N. Xiong, Y. He, Y. Yang, L. T. Yang, and C. Peng, "A self-tuning multicast flow control scheme based on autonomic technology," in *Proc. 2nd IEEE Int. Symp. Dependable, Auton. Secure Comput. (DASC)*, Indiana Univ., Purdue Univ., Indianapolis, IN, Sep. 29–Oct. 1, 2006, pp. 219–226.
[18] S. Shi and M. Waldvogel, "A rate-based end-to-end multicast congestion control protocol," in *Proc. 5th IEEE Symp. Comput. Commun. (ISCC)*, Antibes-Juan les Pins, France, 2000, pp. 678–686.
[19] N. Xiong, Y. He, and Y. Yang, "An efficient flow control algorithm for multi-rate multicast networks," in *Proc. IEEE Int. Workshop IP Oper. Manag. (IPOM)*, Beijing, China, Oct. 11–13, 2004, pp. 69–76.
[20] L. Tan, N. Xiong, and Y. Yang, "A PGM-based single-rate multicast congestion control scheme," *J. Softw.*, vol. 15, no. 10, pp. 1538–1546, Oct. 2004.
[21] L. Lao, J. H. Cui, M. Gerla, and D. Maggiorini, "A comparative study of multicast protocols: Top, bottom, or in the middle," presented at the IEEE INFOCOM Conf., Barcelona, Spain, Apr. 23–29, 2006.
[22] S. Ratnasamy, A. Ermolinskiy, and S. Shenker, "Revisiting IP multicast," in *Proc. ACM SIGCOMM*, Pisa, Italy, Sep. 11–15, 2006, pp. 15–26.
[23] L. Rizzo, "Pgmcc: A TCP-friendly single-rate multicast congestion control scheme," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, 2000, pp. 17–28.
[24] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 27–31, 2001, pp. 275–286.
[25] N. Xiong, X. Defago, X. Jia, Y. Yang, and Y. He, "Design and analysis of a self-tuning proportional and integral controller for active queue management routers to support TCP flows," presented at the IEEE INFOCOM Conf., Barcelona, Spain, Apr. 23–29, 2006.



Naixue Xiong (S'07–M'08) received the Ph.D. degrees in computer science from Wuhan University, Wuhan, China and Japan Advanced Institute of Science and Technology, Nomi, Japan, respectively.

He is currently a Research Scientist in the Department of Computer Science, Georgia State University, Atlanta. His research interests include communication protocols, network architecture and design, and optimization theory.

Athanasios V. Vasilakos (M'95–SM'09) is currently a Professor in the Department of Computer and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece, and a Visiting Professor in the Graduate Programme of the Department of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece. He has authored or coauthored more than 200 technical papers in major international journals and conferences. He is the author or coauthor of five books and 20 book chapters in the areas of communications.

Prof. Vasilakos was the General Chair, the TPC Chair, and the Symposium Chair for many international conferences. He was or has been the Editor or/and Guest Editor for many technical journals, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and *ACM Transactions on Autonomous and Adaptive Systems*. He is the Founding Editor-in-Chief of the journals: *International Journal of Adaptive and Autonomous Communications Systems (IJAACS)* and *International Journal of Arts and Technolog (IJART)*. He is the Chairman of the Intelligent Systems Applications Technical Committee (ISATC) of the IEEE Computational Intelligence Society (CIS).

Laurence T. Yang (M'97) received the B.Sc. and B.E. degrees from Tsinghua University, China, and the Ph.D. degree from the University of Victoria, Canada.

Currently, he is a Professor at the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include high performance, embedded and ubiquitous/pervasive computing. He has published around 300 papers (including around 100 international journal papers such as IEEE and ACM Transactions) in refereed journals, conference proceedings and book chapters in these areas. He has been involved in more than 100 conferences and workshops as a Program/General/Steering Conference Chair and more than 300 conference and workshops as a Program Committee Member. He served as the Vice-Chair of IEEE Technical Committee of Supercomputing Applications (TCSA) until 2004, currently is the Chair of IEEE Technical Committee of Scalable Computing (TCSC), the Chair of IEEE Task force on Ubiquitous Computing and Intelligence. He is also in the Steering Committee of IEEE/ACM Supercomputing (SC-XY) conference series, and the National Computing Resource Allocation Committee of Canada. In addition, he is the editors-in-chief of several international journals and few book series. He is serving as an editor for around 20 international journals. He has been acting as an author/co-author or an editor/co-editor of 25 books from Kluwer, Springer, Nova Science, American Scientific Publishers and John Wiley & Sons.

Dr. Yang received several Best Paper Awards, Distinguished Achievement Award, 2005; IEEE Distinguished Achievement Award, 2009; Canada Foundation for Innovation Award, 2003; He has been invited to give around 20 keynote talks at various international conferences and symposia.

Yi Pan (S'90–M'91–SM'97) received the B.Eng. and M.Eng. degrees in computer engineering from Tsinghua University, Beijing, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science from the University of Pittsburgh, Pittsburgh, PA, in 1991.

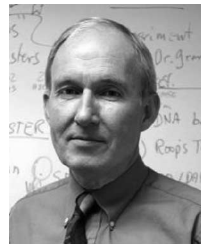
He is currently the Chair and a Professor in the Department of Computer Science and a Professor in the Department of Computer Information Systems at Georgia State University, Atlanta. His research interests include parallel and distributed computing, wireless networks, and bioinformatics. He is the author or coauthor of more than 100 journal papers with more than 40 papers published in various IEEE journals. In addition, he is the author or coauthor of more than 100 papers published in refereed conferences [including International Parallel and Distributed Processing Symposium (IPDPS), the International Conference on Distributed Computing Systems (ICDCS), and INFOCOM]. He has also coedited 30 books (including proceedings) and contributed several book chapters. He is a Co-Inventor of three U.S. patents (pending) and five provisional patents.

Dr. Pan has received many awards from agencies such as NSF, AFOSR, JSPS, IISF, and Mellon Foundation. His recent research has been supported by NSF, NIH, NSFC, AFOSR, AFRL, JSPS, IISF, and the states of Georgia and Ohio. He was an Editor-in-Chief or Editorial Board Member for 15 journals including 5 IEEE Transactions and a Guest Editor for 10 journals. He has served as a PC member for several major international conferences such as INFOCOM, ICC, and IPDPS. He is listed in Men of Achievement, Who's Who in Midwest, Who's Who in America, Who's Who in American Education, Who's Who in Computational Science and Engineering, and Who's Who of Asian Americans.

Cheng-Xiang Wang (S'01–M'05–SM'08) received the B.Sc. and M.Eng. degrees from Shandong University, Shandong, China, in 1997 and 2000, respectively, and the Ph.D. degree in wireless communications from Aalborg University, Aalborg, Denmark, in 2004.

He joined Heriot-Watt University, Edinburgh, U.K., as a Lecturer, in 2005 and became a Reader in 2009. He was a Research Fellow at the University of Agder, Grimstad, Norway, from 2001 to 2005, and a Visiting Researcher at Siemens AG-Mobile Phones, Munich, Germany, in 2004. His current research interests include cognitive radio networks, green/cooperative communications, MIMO, OFDM, UWB, and 4G and beyond. He is the author or coauthor of one book chapter and more than 120 papers published in refereed journals and conference proceedings.

Dr Wang has been an Editor or Guest Editor for several journals including IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He is an Honorary Fellow of the University of Edinburgh.



Art Vandenberg (M'04) received the B.A. degree in English literature from Swarthmore College, Swarthmore, PA, in 1972, the M.V.A. degree in painting and drawing form Georgia State University, Atlanta, in 1979, and the M.S. degree in information and computer systems from Georgia Institute of Technology, Atlanta, in 1985.

He is currently with the Information Systems and Technology Department, Georgia State University, with functional expertise in research computing technology. He is the Chair of the SURAgrid Governance Committee, a cyberinfrastructure initiative of the Southeastern Universities Research Association. His research interests include information integration, resolving metadata heterogeneity, and distributed and cooperative computing solutions

Mr. Vandenberg is Georgia State University's liaison with the IBM Cloud Academy, a global forum for educators, researchers, and information technologists pursuing cloud-computing initiatives. He is a professional member of the Association of Computing Machinery.