Design of a Distributed Flow Control Scheme based on Wireless Multi-rate Multicast Networks

Naixue Xiong[1]
[1]Dept. of Computer Science,
Georgia State University, USA.
Email: {nxiong, pan}@cs.gsu.edu

Xiaohua Jia
Dept. of Computer Science,
City University of Hong Kong.
Email: csjia@cityu.edu.hk

Y. Richard Yang[2], Athanasios V. Vasilakos[3]
[2]Laboratory of Networked Systems, Yale Univ., USA.
[3]Dept. of Computer and Telecomm. Engin.,
University of Western Macedonia, Greece.
Email: yry@cs.yale.edu, vasilako@ath.forthnet.gr

Yi Pan[1], Chin-Chen Chang[4]
[4]Dept. of Information Engineering
and Computer Science,
Feng Chia University, Seatwen, Taiwan.
Email: ccc@cs.ccu.edu.tw

***Abstract*—With the ever-increasing wireless multicast data applications recently, considerable efforts have focused on the design of self-adaptive flow control schemes for wireless multicast service. This paper proposes a novel and efficient distributed flow control scheme for wireless multi-rate multicast (MR-M), based on the well-known Proportional Integral and Derivative (PID) controllers. The PID controller at each router computes its expected incoming rate and feedbacks this rate to its upstream router, such that the local buffer occupancy can be stabilized at an appropriate value. We give the theoretical analysis of the proposed PID controller in terms of system stability. The proposed MR-M controller achieves the fairness in two aspects: 1) The intra-session fairness, i.e., the receivers from the same source within the same multicast session, if they subscribe networks with different capacities, can receive data at different rates; 2) The inter-session fairness, i.e., the link bandwidth is fairly shared among multiple multicast sessions from different sources. Extensive simulations have been conducted and the results have demonstrated a superior performance of the proposed scheme based on system stability, high link utilization, high throughput.**

***Index Terms*—Explicit rate, fairness, flow control, wireless multi-rate multicast, stability.**

I. Introduction

With the ever-increasing wireless multicast data applications recently, considerable efforts have focused on the design of flow control schemes for Multicast congestion avoidance.

There are generally two types of wireless multicast rate control schemes: Single-Rate Multicast (SR-M) and Multi-Rate Multicast (MR-M) [1-2]. The SR-M is not fair to those receivers who are connected to high speed networks and are able to receive data at higher rates. Due to the diverse characteristics and requirements of receivers within a multicast group, it is desirable to have multicast sessions in which different receivers receive data at different rates. This is achieved by MR-M, where the source is able to transmit data to all receivers at different rates that suits the capacity of each individual receiver. Since in MR-M the capacities of network links to different receivers differ and traffic should be accordingly adjusted at the links with different capacities, flow control becomes a very challenging issue. For simplicity, we use multicast to refer wireless MR-M for the rest of the paper, unless otherwise specified.

Several multicast flow control schemes have been proposed in the literature [2-4]. Most researches concern about the bandwidth allocation along the paths from the source to all the destinations. The flow rate is adjusted by some interesting heuristics in IP multicast [5-6], and optimal resource allocation issues are addressed in overlay multicast [7-8]. The max-min fairness is the main objective in this resource allocation [2, 4]. Sarka et al developed a mathematical framework in [4] to model the fair allocation of bandwidth in multicast networks with minimum and maximum rate constraints and presented a distributed algorithm to compute max-min fair rates for various source-destination pairs. As the continuation of the work in [4], the authors presented a unified framework for diverse fairness objectives in [3] and proposed some rate allocation algorithms that maximize the total receiver utility for multirate multicast sessions. However, Lee et al. pointed out in [1] the inadequacies of these resource allocation algorithms: lack of scalability, instability of network traffic, and possible feedback explosion problem. Lee et al., thus, proposed a fair rate computing method for multicast flow control based on the control theory:

$$\alpha_i(n) = -C_P(q_i(n) - \bar{q}_i) - C_I \sum_{m=0}^{n} (q_i(m) - \bar{q}_i), \qquad (1)$$

where $\alpha_i(n)$ is the injecting rate to link i at time slot n, C_P and C_I are the proportional and the integral control gains respectively, and $q_i(n)$ and $\bar{q}_i$ are the buffer occupancy and the target buffer occupancy of the link buffer i respectively. However, this scheme has several shortcomings: 1) The proportional term of the fair rate function only considered the difference between the current and target buffer occupancy, without considering the history of fair rates in a round-trip time and the buffer changes in the consecutive time slots. This makes the system fluctuate to one-time noises and unreliable; 2) The integral term of the fair rate function included the entire historical differences between the actual and target buffer occupancy. As time goes on, the integral component in the fair rate function becomes heavier and heavier, which makes the system difficult to choose the proper value of C_I and to preserve the system stability; 3) It did not consider the situation that may have multiple multicast sessions co-existing in the system (i.e., the inter-session flow control). 4) It gave the stable range for partial control parameters, and it was absent for how to choose the important parameter $|Q_j|$ (the cardinality of the set of locally bottlenecked sessions at link j), this could make the system unstable. To the above shortcomings, this paper proposes an efficient distributed multicast flow control scheme based on the proportional, integral and derivative (PID) controller. This scheme avoids the feedback explosion that often occurs at the routers close to the source by letting each router only feedback to its direct downstream router, not all the way to the source.

II. THE PID FLOW CONTROL SCHEME

The System Model and Notations There is a multicast tree associated with each multicast session. The tree can be established at the connection establishment time, and it can be changed dynamically due to the multicast membership change, network topology change or network congestions. A rate controllers is located at every router (the branch point in the multicast tree). The outgoing links of a router towards its downstream routers or end-users can have different bandwidth capacities and it should send data to its downstream routers or end-users at the rate that suits each individual receiver. This multirate multicast technology has become feasible for multicasting multimedia files due to the advent of efficient fine-grained multimedia transcoding techniques, such as MPEG-4 FGS video standard [6]. Following this standard, it is feasible to freely adjust the video rate to an arbitrary value in real time without time-consuming decoding and re-encoding operations, as long as the target rate is greater than or equal to that of the base layer [1].

The basic idea of flow control of MR-M is as follows. Initially, the multicast source sends data including a forward control packet (FCP) to the destinations along the multicast tree. At each router in the multicast tree, it computes an expected incoming data rate according to its local buffer occupancy, and constructs a backward control packet (BCP) by including this expected incoming rate in the BCP. This BCP is then sent to upstream router. When a router receives a BCP from a downstream router, it will adjust the sending data rate to the expected rate of the downstream router. In this way, as the data and FCP are sent from the source to the destinations, the BCPs are fed back from destinations along the multicast tree all the way to the source. At each router, the sending rate to each of its downstream router is re-adjusted based on the previously received BCP. Eventually the system will reach a stable state: Each router (including the source) sends the data in a rate that is the most desirable to its downstream routers.

To explain our flow control scheme in detail, we assume that there are *m* multicast sources $(S_1, S_2, ..., S_m)$ and one end-to-end CBR (Constant Bit Rate, it is a un-control flow.) source, as shown in Fig. 1. Each multicast source has a multicast session (virtual tree). Each router is connected to some downstream routers and some end-users directly. Fig. 1 is a typical setting for web servers that provide multimedia services, such as VoD (video on demand) or live video broadcast. In the following, we first discuss the flow control within a multicast session. Then, we discuss the inter-session flow control in Section III.

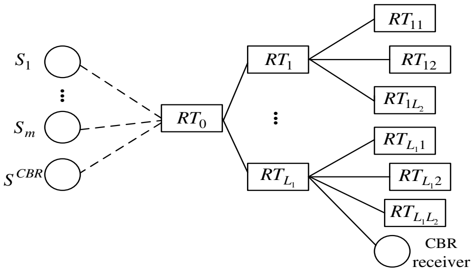

Fig. 1. A multi-rate multicast model, where every router can have downstream routers or end-users.

We now study the buffer occupancy at router RT_i and the expected incoming data rate to RT_i. Suppose RT_i' is the upstream router of RT_i, as shown in Fig. 2. We first introduce the following notations:

x_i: buffer occupancy of RT_i;
$\bar{x}_i$: target buffer occupancy of RT_i;
τ_i: delay from RT_i' to RT_i;
τ_i^*: round-trip delay between RT_i' and RT_i;
I_i: expected incoming rate to RT_i;
O_i: maximum outgoing rate of RT_i to downstream routers and end-users.

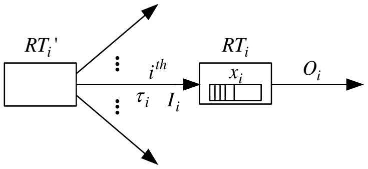

Fig. 2. Router RT_i, its incoming data rate from its upstream router RT_i' and its outgoing data rate.

We assume the real continuous time t in the system is allotted into n time-slots by the sampling period T, here a sampling fine-grain for T is enough to make n an integer. Then we have $t = nT$. Thus, τ_i and τ_i^* are integers. While, based on different sampling fine-grains, there are many different choices for T. A smaller sampling period T makes higher sampling frequency. Actually, the sampling period T little affects the system performance because of the same computing method.

The Buffer Occupancy and the PID Controller The buffer occupancy of RT_i can be determined by the following (see Fig. 2):

$$x_i(n+1) = x_i(n) + I_i(n - \tau_i^*) - O_i(n). \tag{2}$$

Eq. (2) describes the changes of buffer occupancy [9]. It states that the buffer occupancy at time slot $(n+1)$ equals that at time n plus the current incoming rate from the upstream and minus the outgoing rate at time n. $I_i(n - \tau_i^*)$ in Eq. (2) is the actual rate by RT_i computed τ_i^* ago, this is because it takes τ_i for RT_i's upstream router to receive the actual data rate from RT_i and another τ_i for the adjusted data rate to arrive at RT_i (i.e., τ_i^* in total). We assume the upstream router adjust its sending rate immediately after receiving a BCP from a downstream router. The outgoing rate in Eq. (2), i.e., O_i, is the maximum outgoing rate to all downstream routers, because the traffic is multicasted to downstream links (which may have different bandwidth capacities) and the buffer is emptied at the speed of the fastest link.

To stabilize the network system, we must stabilize the buffer occupancy at a desirable level [9] first. The PID rate controller at each router is used to control the buffer occupancy by letting the upstream router to adjust the incoming data rate via a feedback control mechanism. When a router receives the BCPs from its downstream routers, it adjusts its sending rates to the downstream routers to their expected rates, respectively. It then computes its own expected incoming rate based on its targeted buffer occupancy, the current buffer occupancy, and the maximum outgoing rate, and sends a BCP back to its upstream router. The key component of the flow control algorithm is the computation of the expected incoming rate of each router. For router RT_i, $x_i(n)$ that is too high often leads to buffer overflow; a too low $x_i(n)$ causes the bandwidth being underutilized and results in low throughput. We propose a PID rate controller to compute the expected incoming rate for RT_i:

$$\begin{aligned} I_i(n) &= I_i(0) + a(x_i(n) - \bar{x}_i(n)) \\ &\quad + \sum_{t=1}^{\tau_i^*} b_t I_i(n-t) \qquad (3) \\ &\quad + c(x_i(n) - x_i(n-1)), \end{aligned}$$

where $I_i(0)$ is the initial incoming rate to RT_i and $\bar{x}_i(n)$ is the target buffer occupancy at time slot n. Eq. (3) is a typical PID controller, where the 2nd, 3rd, and 4th terms in Eq. (3) are proportional, integral and derivative components, respectively; and a, b_t $(t = 1, 2, ..., \tau_i^*)$ and c are the proportional, integral and derivative parameters, respectively. They will be determined later by the stability criteria. Note that the 3rd term in Eq. (3) (i.e., the integral component) considers the history of expected data rate during a round-trip delay time. This round-trip time is quite significant, compared with the high speed packet switching, in wide area networks. Failure to take this component into consideration, the incoming data rate to a router is based on stale control information, which may result in oscillation of buffer occupancy and degrade the network performance.

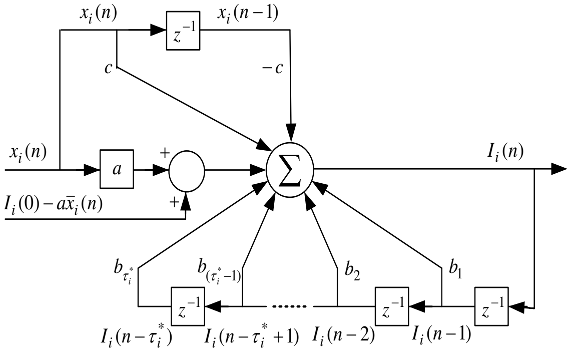

Fig. 3. A recursive digital filter for Multicast control.

In Eq. (3), $(x_i(n) - \bar{x}_i(n))$ is the error signal between the buffer occupancy and the target buffer occupancy at time slot n; $\sum_{t=1}^{\tau_i^*} I_i(n-t)$ is the sum of RT_i's history sending rates during the last round-trip time τ_i^*; $(x_i(n) - x_i(n-1))$ is the differential signal of the buffer occupancy between the current time slot and the last slot. Based on these factors, RT_i computes the incoming rate I_i. And by using the PID controller, the buffer occupancy at a router can be stabilized near its target value.

The recursive Function (3) is a feedback control process, as depicted in Fig. 3, where the lower part is the integral component $\sum_{t=1}^{\tau_i^*} b_t I_i(n-t)$, and the upper part is the derivative component $c(x_i(n) - x_i(n-1))$. The two inputs from the left-hand side in the middle are the proportional component and the initial incoming rate, respectively. Seen from the figure, this is a dynamic controller. The system can stabilize itself based on the internal feedbacks and parameters.

The PID Parameter Setting and Stability Analysis In this section, we analyze the stability of the proposed PID flow control scheme. We apply z-transformation to Eq. (2) and obtain:

$$X_i(z) = \frac{1}{(z-1)}[z^{-\tau_i^*} I_i(z) - O_i(z)], \qquad (4)$$

where $X_i(z)$ and $I_i(z)$ are respectively the z-transforms of $x_i(n)$ and $I_i(n)$: $X_i(z) = \sum_{n=0}^{+\infty} x_i(n) z^{-n}$, and $I_i(z) = \sum_{n=0}^{+\infty} I_i(n) z^{-n}$. Taking the z-transform of Eq. (3), one yields

$$\begin{aligned} I_i(z) &= I_i(0)D(z) + a[X_i(z) - \bar{X}_i(z)] \\ &\quad + \sum_{t=1}^{\tau_i^*} b_t z^{-t} I_i(z) \qquad (5) \\ &\quad + c[X_i(z) - z^{-1} X_i(z)], \end{aligned}$$

where $D(z) = \sum_{n=0}^{+\infty} z^{-n} = \frac{z}{z-1}$, and $\bar{X}_i(z)$ is the z-transform of $\bar{x}_i(n)$. Substitute $X_i(z)$ of Function (4) into Function (5) and obtain:

$$\begin{aligned} (z-1)I_i(z) &= (z-1)I_i(0)D(z) \\ &\quad + a[z^{-\tau_i^*} I_i(z) - O_i(z)] - a\bar{X}_i(z)(z-1) \\ &\quad + \sum_{t=1}^{\tau_i^*} b_t z^{-t} I_i(z)(z-1) \\ &\quad + c(1 - z^{-1})[z^{-\tau_i^*} I_i(z) - O_i(z)]. \qquad (6) \end{aligned}$$

Then

$$\begin{aligned} & I_i(z)[(z-1) - az^{-\tau_i^*} - (z-1)\sum_{t=1}^{\tau_i^*} b_t z^{-t} \\ & \quad - c(1 - z^{-1}) z^{-\tau_i^*}] \\ &= (z-1)I_i(0)D(z) - a\bar{X}_i(z)(z-1) \qquad (7) \\ & \quad - aO_i(z) - cO_i(z)(1 - z^{-1}). \end{aligned}$$

Eq. (7) is the *z-domain representation* of the time variation discrete-time system defined by the Eqs. (2) and (3). Eq. (7) can be rewritten as:

$$\begin{aligned} \Delta(z) \cdot I_i(z) = (z-1)I_i(0)D(z) - a\bar{X}_i(z)(z-1) \\ - aO_i(z) - cO_i(z)(1 - z^{-1}), \end{aligned}$$

by denoting the coefficient of $I_i(z)$ by $\Delta(z)$. That is:

$$\Delta(z) = (z-1)(1 - \sum_{t=1}^{\tau_i^*} b_t z^{-t}) - z^{-\tau_i^*}(c - cz^{-1} + a). \qquad (8)$$

The component $\Delta(z)$ is the *Characteristic Polynomial* (CP) [10] of the multicast closed-loop system given by Eqs. (2) and (3). The coefficients a, b_t $(t = 1, 2, ..., \tau_i^*)$ and c are determined by the stability criteria of control theory. CP (8) is closely related to the stability of the closed-loop system [10].

Due to the space limitation, the proof of system stability is not shown here. We conclude that the CP is stable if $\varepsilon < 1/(\tau_i^* + 2)$. Here, we have:

$$a = (\tau_i^* + 2), b_t = t\varepsilon - 1, \texttt{and} \ c = -\varepsilon, \tag{9}$$

Thus, when $\varepsilon < 1/(\tau_i^* + 2)$, all the zeros of Eq. (8) lie within the unit disk, and the network system (2) with the controller (3) is stable.

III. IMPLEMENTATION OF PID CONTROLLER

Here fairness implies the share of limited resources (buffer occupancy at routers and link bandwidth).

Intra-session Fairness Algorithm As mentioned above, MR-M can allocate different rates to different users, which ensures good intra-session fairness.

The MR-M flow control mechanism works as follows. Initially, the multicast source sends data at an initial rate, together with FCP to its downstream routers. Upon receiving a BCP from a downstream router, it adjusts the sending rate to the downstream router according to its expected incoming rate embedded in the BCP.

When a router in the multicast tree receives data and FCP, it replicates data and FCP to outgoing links to its downstream routers (or end-users) in accordance with their expected rates respectively. After receiving a BCP from a downstream router, it adjusts the sending rate to the downstream router correspondingly. Since the sending rate is changed (i.e., the maximal outgoing rate of the buffer in Eq. (2) may be changed), it needs to re-calculate the expected incoming rate and feeds back this information to its upstream router in a BCP. The BCP, sometimes, also includes information about node joining or leaving the multicast session. The router can adapt to the dynamic changes of the multicast group. Moreover, a router also uses BCPs and FCPs to measure the round-trip delay to its upstream router [11].

An end-user is active to send the expected receiving rate to the upstream router, and processes the data and control packets in a similar way as a router does, except that it does not forward out any data. In such a way, the data rate in the whole multicast tree eventually reaches stability through the PID controllers' continuous adjustment.

The proposed intra-session flow control mechanism has the following features: 1) It is fully distributed. Each router computes its expected incoming data rate based on its local available information. There is no need of global knowledge about the whole network and the multicast group members. 2) It is fair to each receiver. The data rate received by each receiver depends on the capacity of the network it subscribes. A receiver connected to a fast network will not be affected by multicast group members that are connected to slow networks. 3) The buffer occupancy can be stabilized quickly, which is resilient to one-time fluctuation traffics. This is because in our flow control method, we take into consideration the change within the round-trip time.

The Inter-session Fairness Algorithm In this section, we discuss the inter-session fairness, which means different multicast sessions share the link bandwidth in a fair fashion.

Suppose there are m multicast sessions and one CBR connection going through router RT_i (it can be multiple or zero CBR connections). We assume that the CBR has higher priority than the multicast flows. Each multicast session has a buffer in RT_i to control the data rate of this session. The buffer size for a multicast session can be proportional to its maximal outgoing rate from RT_i. Let I_i^j denote the expected incoming rate of Session j ($1 \leq j \leq m$) to RT_i and I_i^{CBR} denote the data rate of the CBR connection. Each I_i^j is computed by Eq. (3) according to the buffer occupancy of Session j. Suppose the bandwidth capacity of the link from its upstream router to RT_i is l_i. If l_i is large enough to support all multicast sessions and the CBR traffics, RT_i can receive data for all sessions according to their expected rates; otherwise, we reduce the traffic of all sessions in the same proportion. We here introduce a ratio w_i: $w_i(n) = (l_i - I_i^{CBR})/(\sum_{j=1}^m I_i^j(n))$.

If $(l_i - I_i^{CBR}) < (\sum_{j=1}^m I_i^j(n))$, the actual expected incoming rate of Session j, denoted by $I_i'^j$, will be

$$I_i'^j(n) = I_i^j(n) \cdot w_i. \tag{10}$$

The value of $I_i'^j$ is run back to the upstream router of RT_i as the sending rate for Session j. By doing so, RT_i allocates bandwidth to all sessions in the same proportion to their desired traffic volume without exceeding the bandwidth capacity of the incoming link.

In addition to the distributed nature and fast convergence of the method, this inter-session flow control mechanism has the following features: 1) It is fair to all multicast sessions. The link bandwidth is shared by all multicast sessions in the same proportion to their originally expected data rate, which avoids starvation of the sessions that require less bandwidth. 2) It has high bandwidth utilization. The bandwidth of the links is fully utilized by all multicast sessions. 3) The mechanism is simple and highly efficient. The expected data rates for all multicast sessions are computed by the simple formula Eq. (10).

IV. Performance Evaluation

Here we pay more attention to sending rates of sources, buffer occupancy, link utilization, receiving rates of routers and end-users. We assume that the link delay is dominant compared to the processing delay or queuing delay.

Simulation Model The simulation model is shown in Fig. 4. There are two multicast sources S_1 and S_2, and one CBR source S^{CBR}. The delays between any two nodes are shown in Fig. 4.

For convenience, we group together the receivers having similar receiving rates. Thus, we select a single receiver in each group as a representative of the group (see Fig. 4). Here the parameters for Receivers in the simulation model are as follows. The expected receiving rates of Receivers 11, Receivers 12, Receivers 13, Receivers 14-1 and Receivers 14-2 from Source 1 are 2 Mbps, 2 Mbps, 4 Mbps, 5 Mbps and 4 Mbps, respectively. And the expected receiving rates of Receivers 21, Receivers 22, and Receivers 23 from Source 2 are 1.5 Mbps, 4 Mbps and 3 Mbps, respectively.

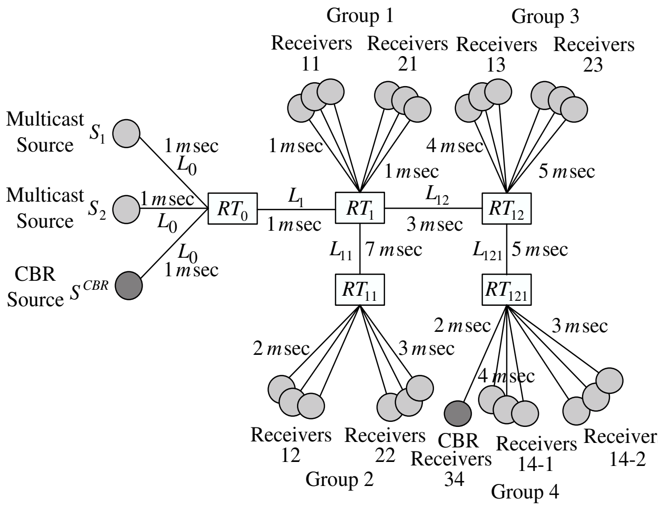

Fig. 4. The simulation model.

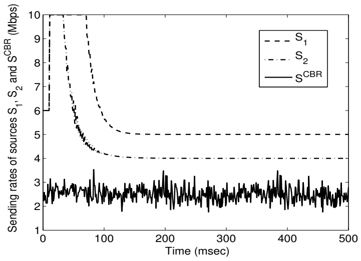

Fig. 5. Sending rates of sources: S_1, S_2, and S^{CBR}.

The buffer sizes of routers RT_0, RT_1, RT_{11}, RT_{12} and RT_{121} are 500 Mb, 450 Mb, 170 Mb, 360 Mb and 120 Mb, respectively. The target buffer occupancy is always set to 90% of the buffer size. The bandwidths of links L_0, L_1, L_{11}, L_{12}, L_{121} are 10 Mbps, 14 Mbps, 6.5 Mbps, 12 Mbps and 9 Mbps, respectively. The initial sending rate of the multicast source is 6 Mbps. For the sending rate of the CBR source, we sampled the incoming traffic of the Japan Advanced Institute of Science and Technology, and collected the statistical data for 30 days until Friday, June 23, 2006 at 6:20 UTC. We, in this case, randomly chose one day's data as the sending rate of the CBR source.

Based on the above simulation parameters and theory in Section II, we get: when $\varepsilon < 1/(\tau_i^* + 2)$, all the poles of Eq. (8) lie within the unit disk, and the original network system (2) with the controller (3) is stable. Here we set $\varepsilon = 1/(\tau_i^* + 3)$ to ensure stability of network system. Then we get the corresponding control parameters a, b_t, c based on Eq. (9). For RT_1, $\tau_1^* = 2$, then $\varepsilon = 1/5$, $a = -1/5$, $c = -1/5$, and $b = [b_1, b_2]$, i.e., $b = [-4/5, -3/5]$. For RT_{11}, $\tau_{11}^* = 14$, then $\varepsilon = 1/17$, $a = -1/17$, $c = -1/17$, and $b = [b_1,\ b_2, b_3, ..., b_{14}]$, i.e., $b = [-16/17,\ -15/17, -14/17, ..., -4/17, -3/17]$. For RT_{12}, $\tau_{12}^* = 6$, then $\varepsilon = 1/9$, $a = -1/9$, $c = -1/9$, and $b = [b_1, b_2, b_3, ..., b_6]$, i.e., $b = [-8/9, -7/9,\ -6/9, ...,\ -4/9, -3/9]$. In group 4, both subgroups 14-1 and 14-2 receive data from the multicast source S_1. However, their delays and expected receiving rates are different. For RT_{121}, the delay from RT_{12} to RT_{121} is $\tau_{121}^* = 10$, then we have $\varepsilon = 1/13$, $a = -1/13$, $c = -1/13$, and $b = [b_1, b_2, b_3,\ ..., b_9, b_{10}]$, i.e., $b = [-12/13, -11/13, -10/13, ..., -4/13, -3/13]$.

Performance Evaluation The simulation results are shown in Figs. 5-13. Fig. 5 shows the sending rates of multicast sources S_1 and S_2, and CBR source S^{CBR}, which corresponding to multicast sessions 1, 2, 3, respectively, in the simulation figures. For multicast sources S_1 and S_2, the sending rates remain at the initial rate 6 Mbps, because the expected rates have not reached the sources due to network delay. When the end-users first receive the data and feedback the BCPs to the routers to which they connect, the PID controllers at the routers start to compute their expected receiving rate and feedback their BCPs to their upstream routers. As time goes on, when the BCPs arrive at the sources, the multicast sources adjust the sending rate gradually to stabilize the traffic at all routers. As we see from Fig. 5, the sending rates of the sources S_1 and S_2 are stabilized at the rates of 5 Mbps at time 130 ms, and 4 Mbps at time 119 ms, respectively.

Figs. 6-9 show the fair rates of Sessions 1, 2 and 3 in L_1, L_{11}, L_{12} and L_{121}, respectively. The fair rate of a session is the actual data rate of a session based on the inter-session fairness flow control method. In Fig. 6, there are some fluctuations at the beginning because of network delay and response of the PID controllers. The PID controllers quickly adjust the expected rates for downstream routers and users. Then the fair rates for Session 1 and Session 2 in L_1 immediately stabilize at 5 Mbps and 4 Mbps at 127 ms and 116 ms, respectively. Figs. 7-9 demonstrate the similar trend as Fig. 6. Although the rates fluctuate at the beginning, they rapidly become stable, and reach the maximal output rate. From the simulation results, we can see that the inter-session fairness on every link is ensured. From Figs. 6-9, we can also find the data flows of a session on different links vary depending on the link bandwidth to which they connect. Thus, our method also ensures good intra-session fairness (see Table I).

The ratios of buffer occupancy to buffer sizes of RT_1, RT_{11}, RT_{12} and RT_{121} are shown in Figs. 10-11. The ratios of RT_1, RT_{11}, RT_{12} and RT_{121} gradually stabilize at 92%, 86%, 95%, and 91% at 121 ms, 120 ms, 98 ms and 95 ms, respectively. Thus, our control method can quickly stabilize the buffer occupancy, very close to the target value. The link utilization of L_1, L_{11}, L_{12} and L_{121} are shown in Figs. 12-13. For link L_{11}, there are only multicast flows, so the link utilization stabilizes at 92% at 121 ms. For links L_1, L_{12}, L_{121}, because the CBR flow passes through them, there are some fluctuations. Nevertheless, the link utilizations are eventually stabilized around 82%, 88%, and 83%, respectively. From the above analysis, our method achieves a good link utilization.

Based on the simulation results, we can see that MR-M scheme presented in this paper has good intra-session and inter-session fairness, good stability, and high link utilization.

How the parameters influence system stability Keep in mind that theory in Section II, any value of ε that meets the condition of $\varepsilon < 1/(\tau_i^* + 2)$ will stabilize the system. This section explores how the different values of ε in stable range influence the performance of system. Then we give a possible rule about how the control parameters influence system stability. It is better for us to choose the control gains, so as to ensure good system performance.

TABLE I
STABLIZED RATES AND TIME FOR REACHING STABILITY.

Parameters	S_1	S_2	Session 1 in L_1	Session 1 in L_{11}	Session 1 in L_{12}	Session 1 in L_{121}	Session 2 in L_1	Session 2 in L_{11}	Session 2 in L_{12}
Stablized Rate (Mbps)	5	4	5	2	5	5	4	4	3
Response time (ms)	130	119	127	91	107	88	116	105	89

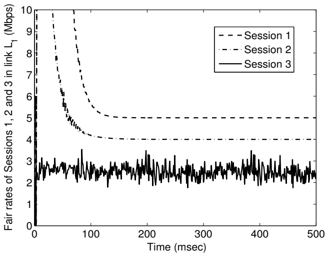

Fig. 6. For Sessions 1-3 in L_1.

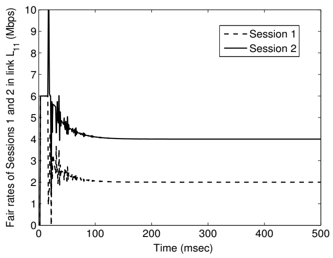

Fig. 7. For Sessions 1-2 in L_{11}.

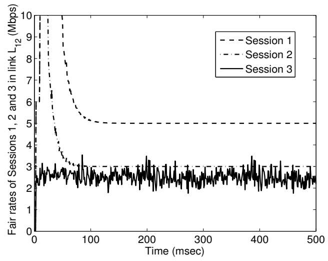

Fig. 8. For Sessions 1-3 in L_{12}.

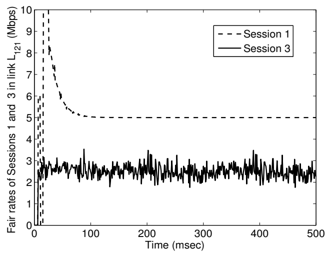

Fig. 9. For Sessions 1,3 in L_{121}.

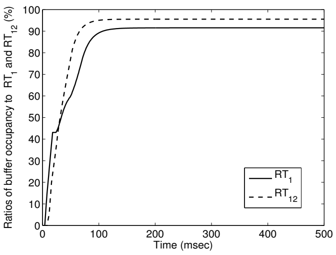

Fig. 10. Buffer occu. to RT_1, RT_{12}.

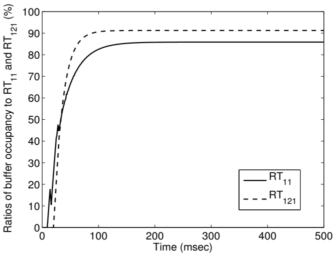

Fig. 11. Buffer occu. to RT_{11}, RT_{121}.

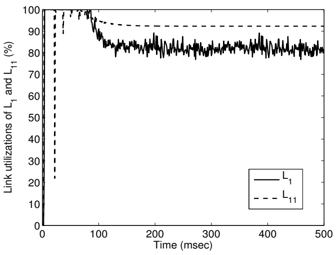

Fig. 12. Utilizations of L_1, L_{11}.

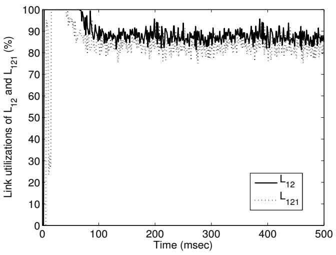

Fig. 13. Utilizations of L_{12}, L_{121}.

Extensive simulations have been conducted with the same simulation settings as above details in Section IV. Here, we randomly choose five typical cases, i.e., $\varepsilon = 1/(\tau_i^* + 6)$, $\varepsilon = 1/(\tau_i^* + 10)$, $\varepsilon = 1/(\tau_i^* + 50)$, $\varepsilon = 1/(\tau_i^* + 100)$, and $\varepsilon = 1/(\tau_i^* + 1000)$. Based on Eq. (9), we can get the corresponding control parameters a, b_t, c, respectively. Due to the limitation, the details on parameters and simulation figures are not shown here.

In summary, we conclude that all the values ε that satisfy $\varepsilon < 1/(\tau_i^* + 2)$ can make system stable. And different values ε in stable range lead to different degrees of system stability. Furthermore, in stable range, the proper values ε near the threshold $\varepsilon = 1/(\tau_i^* + 2)$ may get better system performance than those farther from the threshold based on the above simulation results.

V. CONCLUSIONS

In this paper, we presented an efficient flow control scheme for wireless MR-M, using an explicit rate feedback mechanism. We also proposed a Proportional, Integral, and Derivative (PID) controller to stabilize the buffer occupancy at routers, and the traffics in the network can thus be stabilized. What is more, we used the modern control theory to determine the PID parameters to ensure the stability of the control loop, in terms of sending rate and buffer occupancy. The proposed scheme achieves good intra-session and inter-session fairness, and high link utilization. Simulation results have demonstrated the superior performance of our scheme in terms of system stability, link utilization, and intra-session and inter-session fairness.

REFERENCES

[1] H. W . Lee and J. W. Cho. A distributed max-min flow control algorithm for multi-rate multicast flows. In *Proceedings of IEEE Global Communications Conference*, pages 1140–1146, vol. 2, Dallas, Texas, USA, November 29–December 3, 2004.

[2] S. Sarkar and L. Tassiulas. Fair distributed congestion control in multirate multicast networks. *IEEE/ACM Transactions on Networking*, 13(1):121–133, February 2005.

[3] K. Kar, S. Sarkar, and L. Tassiulas. Optimization based rate control for multirate multicast sessions. In *Proceedings of IEEE INFOCOM 2001*, pages 123–132, Anchorage (Alaska), USA, April 22–26, 2001.

[4] S. Sarkar and L. Tassiulas. Distributed algorithms for computation of fair rates in multirate multicast trees. In *Proceedings of IEEE Infocom 2000*, pages 52–61, Tel Aviv, Israel, March 26–30, 2000.

[5] S. Ratnasamy, A. Ermolinskiy and S. Shenker. Revisiting IP multicast. In *Proceedings of ACM SIGCOMM 2006*, pp. 15–26, Pisa, Italy, Sep. 11–15, 2006.

[6] D. Smith. IP TV bandwidth demand: Multicast and channel surfing. In *Proceedings of IEEE INFOCOM 2007*, Anch., USA, May 6-12, 2007.

[7] M. Afergan and R. Sami. Repeated-game modeling of multicast overlays. In *Proceedings of IEEE INFOCOM 2006*, Barcelona, Apr. 23–29, 2006.

[8] X. Jin. A Cost-based Evaluation of End-to-End Network Measurements in Overlay Multicast. In *Proceedings of IEEE INFOCOM 2007*, Anchorage, USA, May 6-12, 2007.

[9] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks: the case of single congested node. *IEEE/ACM Transaction on Networking*, 1(6):693–708, December 1993.

[10] W. Kamen, B. S. Heck. *Fundamentals of Signals and Systems Using the Web and Matlab*. The 2th edition, prentice Hall, INC, pages 101–105, and pages 581–597, 2002.

[11] J. Widmer and M. Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of ACM SIGCOMM 2001*, pages 275–286, San Diego, California, USA, August 27–31, 2001.