Computer Networks 54 (2010) 1727–1738

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed max–min flow control for multi-rate overlay multicast

Hyang-Won Lee [a,*], Jeong-woo Cho [b], Song Chong [c]

[a] *Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Mass Ave., Cambridge, MA 02139, United States*
[b] *Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway*
[c] *School of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

ARTICLE INFO

Article history:
Received 25 August 2009
Received in revised form 8 January 2010
Accepted 5 February 2010
Available online 12 February 2010
Responsible editor: A. Kshemkalyani

Keywords:
Multi-rate overlay multicast
Max–min flow control
Nyquist stability criterion
Feedback consolidation

ABSTRACT

We present a distributed algorithm to compute bandwidth max–min fair rates in an overlay multicast network supporting multi-rate data delivery. The proposed algorithm is *scalable* in that it does not require each logical link to maintain the saturation status of all sessions and virtual sessions traveling through it, *stable* in that it converges asymptotically to the desired equilibrium satisfying the minimum plus max–min fairness even in the presence of heterogeneous round-trip delays, and has explicit *link buffer control* in that the buffer occupancy of every bottlenecked link in the network asymptotically converges to the pre-defined value. The algorithm is based on PI (proportional integral) control in the feedback control theory and by appealing to the Nyquist stability criterion, a usable stability condition is derived in the presence of sources with heterogeneous round-trip delays. In addition, we propose an efficient feedback consolidation algorithm which is computationally simpler than its hard-synchronization based counterpart and eliminates unnecessary consolidation delay by preventing it from awaiting backward control packets that do not directly contribute to the session rate. Through simulations we further verify the analytical results and the performance of the proposed multi-rate multicast flow control scheme based on these two algorithms.

1. Introduction

This paper deals with a multi-rate multicast flow control problem in service overlay networks. The service overlay network is maintained by the overlay network provider who deploys a number of specially-designed overlay nodes and connects the nodes by purchasing logical links (interchangeably, links) with certain bandwidth guarantee from ISPs (Internet Service Providers) [1,2]. Consequently, the service overlay network can easily support value-added multicast services by implementing additional functionalities at the overlay nodes, which might be prohibitive in IP (Internet Protocol) multicast due to their complexities.

We consider a service overlay network[1] where every link bandwidth is stably provisioned. Suppose that an efficient and scalable multicast routing mechanism exists in the service overlay network. The problem we address in this paper is a multicast flow control problem given a multicast tree pre-determined by the multicast routing mechanism. Note that this overlay network can also be understood as a *virtualization* which was proposed for the flexibility in the future Internet. It enables to support multiple customized protocols on a single physical platform by isolating the resources such as bandwidth, CPU and forwarding table, and thus forming a virtual network for each protocol [3,4].

Multicast flow control schemes can be classified into single-rate schemes and multi-rate schemes according to

[*] Corresponding author. Tel.: +1 617 253 8197.
E-mail addresses: hwlee@mit.edu (H.-W. Lee), jeongwoo@q2s.ntnu.no (Jeong-woo Cho), song@ee.kaist.ac.kr (S. Chong).

[1] Unless otherwise specified, overlay node, overlay link and service overlay network will be simply called node, link and network, respectively.

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.02.004

their way of determining the allowed flow rate of a session on each link in the tree. In a single-rate multicast scheme, the incoming flow rate of a session at every branching point in its tree is enforced to be the minimum of the rates that can be accommodated by its participating branches. The disadvantage of single-rate schemes is that *intra-session fairness* is not guaranteed, meaning that no matter how fast its path rate is, all the receivers must receive data at a single rate which is the slowest path rate. This could be a serious problem in practice, considering, for example, network users who pay more to possess a higher-speed access link [5]. In fact, the heterogeneity of multicast receivers was already observed through a year-long experiment [6].

Multi-rate multicast schemes solve this intra-session unfairness problem at the cost of increasing complexity of branching nodes. In a multi-rate scheme, the incoming flow rate of a session at every branching point in its tree is enforced to be the maximum of the rates that can be accommodated by its participating branches. By doing so, the sending rate at the source will eventually be the maximum of the rates that can be accommodated by the entire paths to individual receivers. Since the source sends data at the maximum path rate, it is necessary to convert down the incoming flow rate at every branching point to the values that can be accommodated by its participating branches. Provided such a rate adaptation functionality at every branching point, each virtual session (VS), defined as each source–receiver pair in a multicast session, will eventually receive data at an independently trimmed rate which is equal to the rate allowed by its entire path. Therefore, multi-rate schemes can ensure intra-session fairness (interchangeably, virtual session fairness) and are desirable for multicast with heterogeneous receivers.

Several multi-rate multicast flow control algorithms have been proposed and analyzed [7–9]. These algorithms differ in their target fairness; namely, they adopt bandwidth max–min fairness, aggregate utility maximization and utility max–min fairness, respectively. The problems with these algorithms are as follows. First, they have lack of scalability since they require each node to keep maintaining the saturation status of every session and VS traveling through it. Second, they have no explicit control over link buffer occupancy so that the allocated rates can wander considerably before converging and thus link flow can exceed the capacity temporarily. Third, no explicit and usable stability condition has been given particularly in the presence of heterogeneous round-trip delays. Last, the feedback explosion problem [10] has not been addressed.

Our work in this paper solves the aforementioned drawbacks of the previous algorithms. We suppose that fine-grained multimedia transcoding techniques are available and feasible to implement at specially-designed overlay nodes. Note that this is also assumed in [7–9], but is no longer merely a supposition today because of the advent of efficient fine-grained scalable coding methods such as SVC (scalable video coding) standard [11]. The SVC standard enables 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 no less than that of the base layer.

We develop a distributed algorithm to compute bandwidth max–min fair rates in a multi-rate overlay multicast network. Our algorithm is *scalable* in that it does not require each node to keep maintaining the saturation status of every session and VS traveling through the node. It is also *stable* in that it converges asymptotically to the desired equilibrium even in the presence of heterogeneous round-trip delays with an explicit and usable stability condition. Further, it has *explicit link buffer control* in that the buffer occupancy of every bottleneck link in the network asymptotically converges to the desired value. In addition, we propose an efficient *soft-synchronization feedback consolidation* algorithm which is computationally simpler but performs better than the hard-synchronization counterpart [12]. The proposed consolidation algorithm eliminates unnecessary consolidation delay by preventing the algorithm from awaiting backward control packets (BCPs) that do not directly contribute to the session rate. Moreover, it limits the number of BCPs traveling through a link in the backward direction to that of forward control packets (FCPs) traveling through it in the forward direction, thereby solving the feedback explosion problem.

2. The algorithm

Fig. 1 depicts the functional block diagram of the proposed multi-rate multicast flow control scheme at a branching node in a overlay multicast network. In the forward direction, a multicast flow branches at the incoming link of the node and is forwarded onto all of its outgoing links by the packet forwarder. Rate adaptors associated with each outgoing link are also located at the incoming link. At each outgoing link, there is a single FIFO queue to multiplex all flows traveling through the outgoing link. The fair rate computation algorithm runs independently at each outgoing link using the occupancy information of the FIFO queue. The source of a multicast session issues and transmits an FCP (forward control packet) in the forward direction repeatedly upon every transmission of F data packets, in order to communicate flow control related information with the nodes in the tree. FCPs are also multicasted as data packets are. The receivers of the multicast session send these control packets back to the source as soon as they receive them. These control packets in the backward direction are BCPs (backward control packet). The feedback consolidation algorithm runs at the incoming link in the backward direction. It merges the BCPs received from different branches into one BCP. We assume in the paper that the forward path and the backward path of each VS are identical and the result of the fair rate computation is written onto BCPs instead of FCPs.

Before we state the algorithm in details, we summarize data structures to be maintained at a branching node in Table 1 and provide the pseudocode of the proposed router and source algorithms in Fig. 2.

2.1. Fair rate computation

The proposed fair rate computation is based on PI control in the feedback control theory [13,14] and has

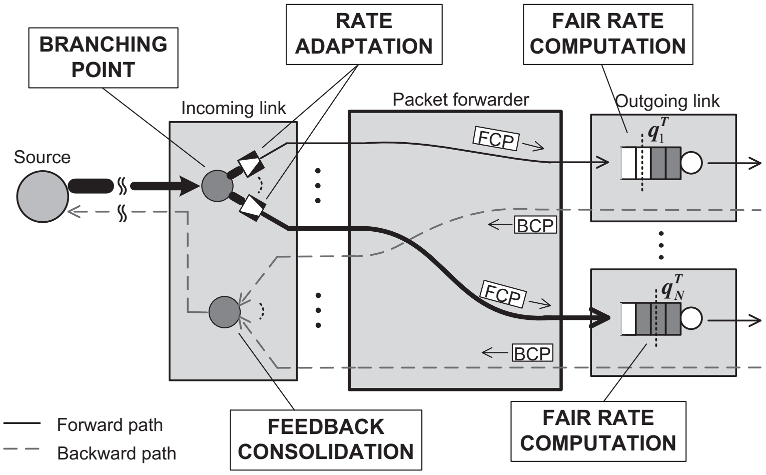


Fig. 1. The functional block diagram at a branching node.

Table 1
Data structures used at a branching node.

Data structures	
Outgoing link j:	$FairRate, ErrorSum$
Incoming link:	$Token[M], MaxBr[M],$ $MaxBrRate[M], BrRate[M][N]$
Source:	SDR, ADR, MDR, PDR
FCP/BCP:	ADR, MDR
	Description
FairRate	Fair rate at outgoing link j
ErrorSum	Cumulative $q_j[k] - q_j^T$
M	Maximum # of multicast sessions
N	Maximum # of branches per session
$Token[i]$	Token for upstream transmission of BCP
$MaxBr[i]$	Session i's branch having maximum rate
$MaxBrRate[i]$	Maximum rate of session i's branches
$BrRate[i][j]$	Rate of session i's branch j
SDR	Current transmission rate at source
ADR	Allowed data rate
MDR	Minimum data rate to be guaranteed
PDR	Peak rate constraint

the following form. For each outgoing link j, its fair rate, $f_j[k]$, is calculated periodically upon every T epoch by

$$f_j[k] = -C_P\left(q_j[k] - q_j^T\right) - C_I \sum_{n=0}^{n=k}\left(q_j[n] - q_j^T\right), \tag{1}$$

where $C_P > 0$ and $C_I > 0$ are the proportional and the integral control gains respectively, $q_j[k]$ is the queue length at the link buffer j, and q_j^T is its target queue length. The choice of C_P and C_I determines the convergence rate of the iteration as well as the stability of the multicast network. In Section 4 we will give the sufficient and necessary condition to ensure stability and the optimal choice of C_P and C_I considering the convergence rate.

In contrast to the previous fair rate allocation algorithms in [7–9], the proposed algorithm in (1) is completely independent of the number of sessions and VSs traveling through the link and thus highly scalable. Moreover, it jointly controls rate allocation and link buffer occupancy, meaning that as the iteration proceeds, it makes the link buffer occupancy converge to the target value, i.e., $\lim_{k\to\infty} q_j[k] = q_j^T$, while finding the max–min fair rate (This will be proved later in Section 4). Such an explicit control of the link buffer occupancy is desirable in practice since without this, the allocated rates can wander considerably before converging and the link flow can exceed the capacity temporarily yielding uncontrolled link buffer occupancy before converging. Furthermore, the link buffer occupancy even in the steady state can be arbitrary and unpredictable.

Suppose that there exists an admission control such that at every link in the network the sum of *MDR*s of all sessions sharing the link is always less than its capacity. Then, the fair rate computation in (1) can be easily extended to support *MDR*. We allocate $f_j[k] + MDR$ to the sessions which require *MDR* guarantee. This makes the so-called *minimum plus max–min fairness* achievable, implying that *MDR*s of all sessions in the network can be guaranteed and whatever the bandwidth remains after the guarantee will be shared by competing sessions in the max–min fair sense.

The major role of BCPs is to inform upstream nodes of the fair rates computed locally by each link in the tree. Consider an outgoing link, say j. Upon receipt of a BCP from its downstream node, the fair rate computed locally by this link, $f_j[k] + MDR$, is compared with the fair rate of its downstream carried by the BCP.*ADR* field of the BCP, and the smaller value is written onto the field and delivered to the upstream. Note that the *MDR* value is also available from the *MDR* field of the BCP (See the first part of Fig. 2).

2.2. Feedback consolidation

Consider branch j of multicast session i. The rate allocated by branch j to session i is stored in $BrRate[i][j]$. Upon receipt of a session i's BCP from the outgoing link j, this value is updated as $BrRate[i][j] =$ BCP.*ADR* since the BCP.*ADR* carries the information on the rate allowed by branch j. Rate adaptation is executed on branch j before forwarding

```
⟨ Overlay node algorithm ⟩

⟨ Outgoing link $j$ ⟩
Upon every T epoch (say, time k)
    $ErrorSum \ += \ q_j[k] - q_j^T$
    $FairRate \ = \ -C_P(q_j[k] - q_j^T) - C_I \cdot ErrorSum$
Upon receipt of a BCP from its downstream
    If $BCP.ADR > FairRate + BCP.MDR$
        $BCP.ADR = FairRate + BCP.MDR$
    Send the BCP to its intended incoming link

⟨ Incoming link ⟩
Upon receipt of a forward packet of session $i$
    If the packet is an FCP
        $Token[i] = 1$
    Make as many copies as intended branches
    Execute rate adaptation on each branch $j$ to enforce $BrRate[i][j]$
    Forward each output to its intended outgoing link
Upon receipt of a BCP of session $i$ from outgoing link $j$
    $BrRate[i][j] = BCP.ADR$
    If $j == MaxBr[i]$
        $MaxBrRate[i] = BCP.ADR$
        If $Token[i] == 1$
            Send the BCP to its upstream node
            $Token[i] = 0$
        Else
            Terminate the BCP
    Else if $BCP.ADR > MaxBrRate[i]$
        $MaxBrRate[i] = BCP.ADR$
        $MaxBr[i] = j$
        If $Token[i] == 1$
            Send the BCP to its upstream node
            $Token[i] = 0$
        Else
            Terminate the BCP
    Else
        If $Token[i] == 1$
            $BCP.ADR = MaxBrRate[i]$
            Send the BCP to its upstream node
            $Token[i] = 0$
        Else
            Terminate the BCP

⟨ Source algorithm ⟩
Transmit data including FCPs at SDR $\leq ADR$
Upon every transmission of F data packets
    Transmit an FCP with
        $FCP.ADR = \infty$
        $FCP.MDR = MDR$
Upon receipt of a BCP
    $ADR = \min\{BCP.ADR, PDR\}$
```

Fig. 2. Pseudocode of overlay node/source algorithms.

data to its intended outgoing link *j* to enforce *BrRate*[*i*][*j*]. On the other hand, if an FCP of session *i* arrives at the incoming link, we set *Token*[*i*] $= 1$. If an arriving BCP of session *i* sees *Token*[*i*] $== 1$, it is eligible to continue to travel through the incoming link in the backward direction. If *Token*[*i*] $== 0$, the BCP must stop traveling. By doing so, the number of session i's BCPs is always restricted to that of session i's FCPs. Therefore, this single-bit token operation solves the feedback explosion problem. If we do not care about feedback explosion, we could write the maximum branch rate onto every BCP received, and send them to the upstream node. In this case, the source will receive the update on the available data rate more frequently. This may result in faster convergence, but at the cost of huge overhead. Note also that in steady state, this will incur only overhead without any benefit.

For consolidation, we use the *locality* information. The key idea in our locality-based consolidation scheme is to cache both ID and rate of a branch which is likely to have the maximum rate among all branches based on history, and to send this cached rate to the upstream node by BCPs. We call this branch as max-branch and store its ID and rate in *MaxBr* and *MaxBrRate* respectively. *MaxBr* and *MaxBrRate* are maintained for each multicast session and updated as follows. Consider an incoming link where session i branches. Suppose that a session i's BCP arrived from outgoing link j. If the link j is the current max-branch of session i, i.e., $j == MaxBr[i]$, then $MaxBrRate[i]$ is updated by the BCP.*ADR* value of this new BCP and $MaxBr[i]$ is kept unchanged, expecting that the link j is still the max-branch. In this case, if $Token[i] == 1$, the BCP is sent to its upstream node with its BCP.*ADR* being unchanged because it is believed to be the one from the max-branch based on locality assumption. On the other hand, if the link j is not the current max-branch of session i and the BCP.*ADR* value of this BCP is greater than the rate of current max-branch of session i, i.e., BCP.$ADR > MaxBrRate[i]$, then both $MaxBr[i]$ and $MaxBrRate[i]$ are updated by j and BCP.*ADR* since it is obvious that the max-branch was changed (See the second part of Fig. 2).

Note that in our scheme, no BCP is waiting for other BCPs for consolidation. Every BCP arriving is processed on the fly and either sent to the upstream node or terminated immediately. Therefore, it completely eliminates the unnecessary consolidation delay to await slow BCPs and thus can improve the transient response, compared to the hard-synchronization based consolidation scheme [12] in which each branching node waits for at least one BCP from all of its branches. Notice that in transient period, our feedback consolidation may incur consolidation error, if a BCP packet is sent to the upstream node while missing the rate information from the actual maximum branch. However, this error will eventually go away unless every BCP from the actual maximum branch is lost, because *MaxBr* and *MaxBrRate* are immediately updated when the new maximum rate is received. This is verified through simulations in Section 5.

2.3. Source algorithm

The source transmits data packets including FCPs at the rate of $SDR\,(\leqslant ADR)$ where *ADR* is updated upon the receipt of a BCP as $ADR = \min\{\text{BCP}.ADR, PDR\}$. It also generates an FCP with FCP.$ADR = \infty$ and FCP.$MDR = MDR$ upon every transmission of F data packets.

As depicted in Fig. 1, our multicast flow control framework has three main functional blocks including *rate adaptation*, *feedback consolidation* and *rate computation*. As seen in Fig. 2, the rate adaption requires $O(MN)$ storage for maintaining branch rates, i.e., $BrRate[M][N]$. Note however that any multi-rate multicast (that adapts the downstream rates with fine granularity) would require this amount of overheads for rate adaptation. The storage needed for our feedback consolidation is $O(M)$ as it additionally uses $token[M], MaxBranch[M]$ and $MaxBranchRate[M]$. It only requires $O(1)$ computation for each BCP arrival as seen in Fig. 2. Note that our feedback consolidation does not require a timeout mechanism not to wait for lost BCPs forever. For rate computation, each outgoing link requires $O(1)$ computation and storage. Overall, we believe that our multicast flow control framework is scalable.

3. Minimum plus max–min fairness

We define minimum plus max–min fairness in multi-rate multicast and establish the bottleneck lemma which will be used to show that our algorithm achieves minimum plus max–min fairness. Let x_k be the rate allocated to VS k and $x = [x_k, k \in V]^T$ where V is the set of all VSs. For $k \in V$, we denote by $E(k)$ the session to which VS k belongs. The minimum rate requirement of VS k is denoted by m_k, and we assume $m_k = m_{k'}$ if $E(k) = E(k')$. We denote the set of virtual sessions passing through link j and belonging to session i by $V(i,j)$, and the set of virtual sessions traversing link j by V_j. Hence, we have $V_j = \bigcup_{i \in S_j} V(i,j)$ where S_j is the set of sessions passing through link j. Let $r_{ij} = \max_{k \in V(i,j)} x_k$, which is the rate of session i at link j.

Definition 1. A rate vector x is said to be *feasible* if it satisfies all the minimum rate requirements and link capacity constraints, i.e., $x_k \geqslant m_k, \forall k$ and $\sum_{i \in S_j} r_{ij} \leqslant \mu_j, \forall j$.

Definition 2. A feasible VS rate vector x^1 is said to be *minimum plus max–min fair* if the following is satisfied: for any other feasible VS rate vector x^2, if there exists VS k such that $x_k^1 - m_k < x_k^2 - m_k$, then there exists VS k' such that $x_{k'}^1 - m_{k'} \leqslant x_k^1 - m_k$ and $x_{k'}^2 - m_{k'} < x_{k'}^1 - m_{k'}$.

Note that the above definition is a straightforward extension of max–min fairness for unicast flows [15] and defined for the rate vector of virtual sessions.

Definition 3. Link j is defined to be a *bottleneck link* of VS k if the following conditions are met:

(1) Link j is fully utilized, i.e., $\sum_{i \in S_j} r_{ij} = \mu_j$.
(2) No other VS k' traversing link j fulfills $x_{k'} - m_{k'} > x_k - m_k$. In other words, $x_k - m_k \geqslant x_{k'} - m_{k'}$, for all $k' \in V_j$.

Definition 4. Session i is said to be *locally bottlenecked* at link j if j is a bottleneck link of one of the virtual sessions belonging to session i.

The following lemma shows the relationship between bottleneck link and minimum plus max–min fairness.

Lemma 1. Bottleneck Lemma *A feasible rate vector is minimum plus max–min fair if every virtual session has a bottleneck link.*

Proof. Let x^1 be a feasible rate vector such that every VS has a bottleneck link, and suppose that there exists VS k such that

$$x_k^1 - m_k < x_k^2 - m_k \tag{2}$$

for any other feasible rate vector x^2. If j is a bottleneck link of VS k, there holds $\sum_{i \in S_j} r_{ij}^1 = \mu_j$, and moreover $\sum_{i \in S_j} r_{ij}^2 \leqslant \mu_j$ by feasibility. As a consequence, there exists a session s such that $r_{sj}^2 < r_{sj}^1$ because $r_{E(k)j}^2 \geqslant x_k^2 > x_k^1 = r_{E(k)j}^1$ where the equality follows from the fact that link j is a bottleneck link of VS k. Let $k' \in V(s,j)$ be a VS of which the rate is its corresponding session's rate at link j, i.e., $x_{k'}^1 = r_{sj}^1$. Then, we can write $x_{k'}^2 \leqslant r_{sj}^2 < r_{sj}^1 = x_{k'}^1$ and thus

$$x_{k'}^2 - m_{k'} < x_{k'}^1 - m_{k'}. \tag{3}$$

By bottleneck condition 2, we have

$$x_k^1 - m_k \geqslant x_{k'}^1 - m_{k'}. \tag{4}$$

Combining (2)–(4) yields $x_{k'}^2 - m_{k'} < x_{k'}^1 - m_{k'} \leqslant x_k^1 - m_k < x_k^2 - m_k$, and this completes the proof. □

4. Analysis

In this section we analyze the steady-state solution as well as the asymptotic stability of a multi-rate multicast network employing the proposed flow control scheme. A usable, sufficient and necessary condition to ensure asymptotic stability of the network is derived in the presence of multicast sessions with arbitrary round-trip delays.

4.1. System model

The system model we consider is depicted in Fig. 3 where we model a single branching node explicitly and the other nodes in the network implicitly for the sake of analytical simplicity. The branching node has M sessions passing through it and N outgoing links with individual FIFO queues. Thus, each session can have at most N branches. We model the system as a continuous-time fluid flow system and this fluid flow model has been widely used for the analysis of network dynamics [16,17]. The detailed assumptions we make for the modeling are as follows:

A.1 The traffic flow and the queueing process are deterministic and continuous in time: This is a typical assumption in the analysis of rate-based congestion/flow control and can be viewed as a deterministic approximation of the underlying stochastic arrival processes [18].
A.2 The round-trip delay τ_i is constant.
A.3 The source always has enough data to transmit at the allocated rate, and sessions do not arrive or leave the network until the system reaches steady state.
A.4 The size of link buffer is infinite: This assumption is only necessary in the analysis and does not affect the practicality of our result, because it can control the buffer occupancy so that it converges to a target value.
A.5 The rate allocated by the downstream nodes of branch j of session i is modeled implicitly as a constant b_{ij}. In fact, the dynamics of b_{ij} are coupled with the dynamics of the branching node. However, for the sake of analytical simplicity, we assume that the other nodes are in steady state and thus b_{ij} is constant.
A.6 The feedback consolidation contains no consolidation delay and errors.

Remark. In Assumptions A.2, A.5 and A.6, we suppose that there exists a neighborhood of the equilibrium point, in which the network becomes fairly close to static. That is, the delays can be viewed as constant (A.2), the node-to-node dynamics are decoupled (A.5), and the max-branch does not change (A.6). By assuming these, we can only find a local stability condition around the equilibrium point, but the condition is fairly strong in that it guarantees the stability with heterogeneous delays. The analysis seems to be intractable without these assumptions. Hence, we investigate the global stability only through simulations, and conclude that such a local stability condition also guarantees the global stability.

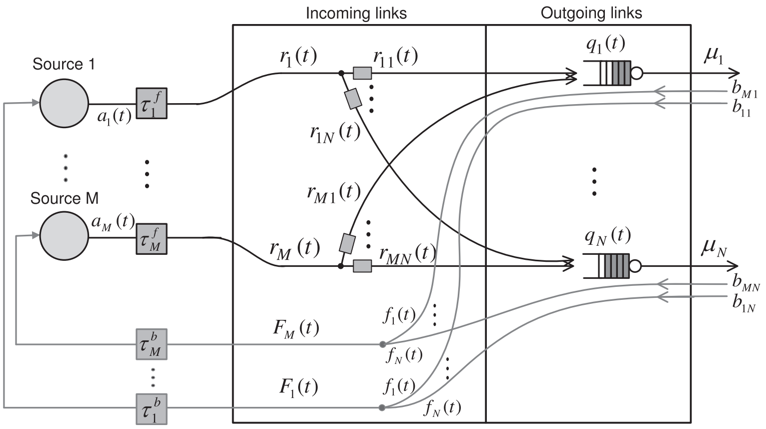


Fig. 3. The system model.

Let $q_j(t)$ be the queue length of link buffer j. By neglecting the buffer floor, the dynamics of the link buffer at each outgoing link j is modeled in continuous time by

$$\dot{q}_j(t) = \sum_{i \in S_j} r_{ij}(t) - \mu_j, \quad j \in L, \tag{5}$$

where $r_{ij}(t)$ is the transmission rate of session i to link j, μ_j is the capacity of link j and L the set of all outgoing links at the branching node $(N = |L|)$. The fair rate computation at link j in (1) can be rewritten in continuous time by

$$f_j(t) = -C_P\{q_j(t) - q_j^T\} - C_I \int_0^t \{q_j(t) - q_j^T\}\,\mathrm{d}t, \quad j \in L, \tag{6}$$

where $f_j(t)$ is the fair rate computed at link j, q_j^T is the target queue length of link buffer j and L is the set of all outgoing links at the branching node. Let m_i be the minimum rate constraint (*MDR*) of session i. In order to make the minimum plus max–min fairness achievable, $f_j(t) + m_i$ is allocated to the session i traversing link j. This rate is compared with the fair rate b_{ij} allocated by the downstream nodes of branch j of session i, to choose the minimum, yielding $\min\{f_j(t) + m_i, b_{ij}\}$.

Let $F_i(t)$ be the fair rate of session i after consolidation. Then, by the assumption A.6, we have

$$F_i(t) = \max_{j \in L_i}[\min\{f_j(t) + m_i, b_{ij}\}], \quad i \in S, \tag{7}$$

where S is the set of sessions passing through the branching node $(M = |S|)$. The consolidated fair rate of session $i, F_i(t)$, is delivered to the source after the backward-path delay τ_i^b of session i. Denote by p_i the peak rate constraint (*PDR*) of session $i(p_i > m_i)$. Then, the source i transmits data at the following rate:

$$a_i(t) = \min[F_i(t - \tau_i^b), p_i], \quad i \in S. \tag{8}$$

Note that $a_i(t) > m_i$ holds since $F_i(t - \tau_i^b) > m_i$ due to the admission control and $p_i > m_i$. The incoming rate $r_i(t)$ of session i at the branching node is the delayed version of source rate $a_i(t)$, i.e., $r_i(t) = a_i(t - \tau_i^f)$ where τ_i^f is the forward-path delay of session i. Let τ_i be the round-trip delay of session i, i.e., $\tau_i^f + \tau_i^b$. Then, it follows

$$r_i(t) = \min[F_i(t - \tau_i), p_i], \quad i \in S. \tag{9}$$

Let L_i be the set of links to which session i branches, then the incoming flow $r_i(t)$ branches out to every $j \in L_i$. The rate adaptor associated with branch j of session i adjusts $r_i(t)$ to match the current fair rate allowed by branch j and its downstream nodes, i.e., $\min[f_j(t) + m_i, b_{ij}]$.

Consider an arbitrary outgoing link j. Let S_j be the set of sessions passing through link j. For $i \notin S_j$, it is obvious that $r_{ij}(t) = 0$. For $i \in S_j$, the rate adaptation is expressed by

$$\begin{aligned} r_{ij}(t) &= \min[r_i(t), f_j(t) + m_i, b_{ij}] \\ &= \min[F_i(t - \tau_i), p_i, f_j(t) + m_i, b_{ij}], \end{aligned} \tag{10}$$

where $F_i(t - \tau_i) = \max_{l \in L_i}[\min\{f_l(t - \tau_i) + m_i, b_{il}\}]$ by (7). Suppose that branch j is the fastest branch of session i in the given network condition. Then, as time goes on, the consolidated fair rate would be determined by branch j, i.e., $F_i(t - \tau_i) = \min[f_j(t - \tau_i) + m_i, b_{ij}]$. On the other hand, suppose that branch $l(l \neq j)$ is the fastest branch of session i in the given network condition. Then, as time goes, the consolidated fair rate would be determined by branch l, i.e., $F_i(t - \tau_i) = \min[f_l(t - \tau_i) + m_i, b_{il}]$, and consequently, $F_i(t - \tau_i) > \min[f_j(t) + m_i, b_{ij}]$ since branch j is not the fastest branch of session i. Therefore, we suppose that as time goes, the system enters a neighborhood of its steady state where (10) can be rewritten as follows. If branch j is the fastest branch of session i in the given network condition,

$$r_{ij}(t) = \min[f_j(t - \tau_i) + m_i, f_j(t) + m_i, b_{ij}, p_i]. \tag{11}$$

Otherwise,

$$r_{ij}(t) = \min[f_j(t) + m_i, b_{ij}, p_i]. \tag{12}$$

Now we define a new variable $d_i(j,t)$ such that if branch j is the fastest branch of session i and $f_j(t) < f_j(t - \tau_i)$, $d_i(j,t) = 0$, else if branch j is the fastest branch of session i and $f_j(t) \geqslant f_j(t - \tau_i), d_i(j,t) = \tau_i$, and else, $d_i(j,t) = 0$. Assuming that the value of $d_i(j,t)$ hardly alternates between 0 and τ_i as the system reaches steady state, we view this variable as a quasi-static variable such that $d_i \leqslant \tau_i$. Then, we can merge (11) and (12) into the following equation in the quasi-static state. For $i \in S_j$,

$$r_{ij}(t) = \min[f_j(t - d_i) + m_i, b_{ij}, p_i], \quad d_i \leqslant \tau_i, \tag{13}$$

whether or not branch j is the fastest branch of session i. Let Q_j be the set of locally bottlenecked sessions at link j. If session i is locally bottlenecked at link j for a given network condition, i.e., $i \in Q_j$, then as time goes, $r_{ij}(t)$ would be determined by the fair rate computed at the link j, i.e., (13) would become $r_{ij}(t) = f_j(t - d_i) + m_i$. If session i is not a locally bottlenecked session of outgoing link j for a given network condition, i.e., $i \in S_j \setminus Q_j$, meaning that the rate of session i is determined either by the peak rate constraint p_i at the source or by the fair rate allocated by the downstream nodes of branch j of session i, b_{ij}, then as time goes, (13) would become $r_{ij}(t) = \min[b_{ij}, p_i]$. Therefore, one can rewrite (13) as

$$r_{ij}(t) = \begin{cases} f_j(t - d_i) + m_i & i \in Q_j, \\ \min[b_{ij}, p_i] & i \in S_j \setminus Q_j. \end{cases} \tag{14}$$

Let us compare the two different expressions for $r_{ij}(t)$ in (10) and (14). The former models the actual interaction between different branches of the same session, whereas the latter neglects it by appealing to a certain technical supposition. Consider an outgoing link j and a session $i \in S_j$. The dynamics of $r_{ij}(t)$ are in fact coupled with the fair rates computed by other branches, as given in (10). However, due to the complex nature of the nonlinearly coupled dynamics between branches, the analysis of global stability considering this coupling could be so involved that in this paper we do not attempt to do it. Instead, we suppose that as time goes, the system enters a certain neighborhood of its steady state where the dynamics of $r_{ij}(t)$ is decoupled with the dynamics of other branches as given in (14). Then, we analyze the local stability of the system in this neighborhood. The global stability considering the coupled dynamics is investigated only through simulations in Section 5.

4.2. Fairness properties

We first analyze the properties of our algorithm, including fairness and buffer occupancy. Suppose that the closed-loop system has an equilibrium point at which the derivatives of the system variables are zero, i.e., $\lim_{t\to\infty}\dot{q}_j(t)=0$ and $\lim_{t\to\infty}\dot{f}_j(t)=0$ for all $j\in L$. Let v^s denote the steady value of any variable $v(t)$, i.e., $v^s=\lim_{t\to\infty}v(t)$. At the equilibrium point, 5, 6, 7, 8 and (14) give us that

$$\sum_{i\in S_j} r_{ij}^s=\mu_j,\quad q_j^s=q_j^T,\quad \forall j\in L, \tag{15}$$

$$F_i^s=\max_{j\in L_i}[f_j^s+m_i,b_{ij}],\quad a_i^s=\min[F_i^s,p_i],\quad \forall i\in S, \tag{16}$$

$$r_{ij}^s=\begin{cases} f_j^s+m_i & i\in Q_j\\ \min[b_{ij},p_i] & i\in S_j\setminus Q_j\end{cases}\quad \forall i\in S,\ \forall j\in L. \tag{17}$$

By combining the first equation in (15) and (17), we obtain

$$\sum_{i\in Q_j} f_j^s+\sum_{i\in Q_j} m_i+\sum_{i\in S_j\setminus Q_j}\min[b_{ij},p_i]=\mu_j,\quad \forall j\in L, \tag{18}$$

which implies that

$$f_j^s=\frac{\mu_j-\sum_{i\in Q_j}m_i-\sum_{i\in S_j\setminus Q_j}\min[b_{ij},p_i]}{|Q_j|},\quad \forall j\in L. \tag{19}$$

By substituting (19) for f_j^s in (17), we obtain that for $\forall i\in S$ and $\forall j\in L$,

$$r_{ij}^s=\begin{cases} \frac{\mu_j-\sum_{i\in Q_j}m_i-\sum_{i\in S_j\setminus Q_j}\min[b_{ij},p_i]}{|Q_j|}+m_i & i\in Q_j\\ \min[b_i^s,p_i] & i\in S_j\setminus Q_j.\end{cases} \tag{20}$$

The following theorem summarizes the result.

Theorem 1. *Provided that* $\sum_{i\in S_j}m_i<\mu_j,\ \forall j\in L,$ *and* $\min[b_{ij},p_i]>m_i,\ \forall i\in S_j\setminus Q_j$, *there exists a unique equilibrium point at which (i) the occupancy of each link buffer is equal to its target value* ($q_j^s=q_j^T,\ \forall j\in L$)*, (ii) the capacity of each link is fully utilized* $\left(\sum_{i\in S_j}r_{ij}^s=\mu_j,\ \forall j\in L\right)$*, (iii) for every multicast session, its MDR is guaranteed at all branches in the tree* ($r_{ij}^s>m_i,\ \forall i\in S,\ \forall j\in L$) *and for every link, its unreserved portion of capacity,* $\mu_j-\sum_{i\in S_j}m_i$*, is shared by all sessions traveling through it in max–min fair sense.*

This theorem implies that a multicast network controlled by the proposed scheme has a unique equilibrium point satisfying the multi-rate allocation ensuring intra-session fairness, desired link buffer occupancy and full utilization of link capacity. Moreover, it achieves max–min fairness as shown below.

Theorem 2. *The proposed algorithm achieves minimum plus max–min fairness at steady state.*

Proof. Let $x^s=[x_k^s,\ k\in V]^T$ be the rate allocation vector under our algorithm where x_k^s is the rate allocated to VS k. First, consider an arbitrary VS k. Suppose that all the links in $\widetilde{L}_k$ are under-utilized where $\widetilde{L}_k$ is the set of links used by VS k. Then, we can easily see that $q_j^s=-\infty$, $\forall j\in\widetilde{L}_k$. According to (6), there holds $f_j^s=\infty,\ \forall j\in\widetilde{L}_k$, and thus, $x_k^s=\infty$ due to $x_k^s=\min_{j\in\widetilde{L}_k}\{f_j^s+m_k\}$. This contradicts the supposition that all the links in $\widetilde{L}_k$ are under-utilized at steady state. Therefore, there exists at least one fully-utilized link on the path of VS k. Note that this argument holds even if we incorporate buffer flooring in (5).

Let $j^*=\arg\min_{j\in\widetilde{L}_k}\{f_j^s\}$, then $x_k^s=f_{j^*}^s+m_k$. Because $x_{k'}^s-m_{k'}=\min_{j\in\widetilde{L}_{k'}}f_j^s,\ \forall k'\in V_{j^*}$, it follows that $x_k^s-m_k=f_{j^*}^s\geqslant x_{k'}^s-m_{k'},\ \forall k'\in V_{j^*}$. Hence, for each VS k, there exists at least one link at which $x_k^s-m_k\geqslant x_{k'}^s-m_{k'}$ for any VS k' traversing the link. It is obvious from the above observation that link j^* is fully utilized because otherwise, $f_{j^*}^s$ will grow to infinity and thus $j^*\neq\arg\min_{j\in\widetilde{L}_k}\{f_j^s\}$. The proof is completed by applying Lemma 1. □

4.3. Asymptotic stability

In this subsection we study the local stability of the closed-loop system in the neighborhood of the equilibrium point where the system is governed by (5), (6) and (14). By appealing to the Nyquist stability criterion [19], the sufficient and necessary condition for the asymptotic stability of the closed-loop system is found in a usable form.

Consider an outgoing link j which has at least one locally-bottlenecked session. By substituting (14) for $r_{ij}(t)$ in (5), we get

$$\dot{q}_j(t)=\sum_{i\in Q_j}f_j(t-d_i)+\underbrace{\sum\nolimits_{i\in Q_j}m_i+\sum\nolimits_{i\in S_j\setminus Q_j}\min[b_{ij},p_i]-\mu_j}_{\text{constant}}, \tag{21}$$

where d_i is either 0 or τ_i as discussed in Section 4.1. The constant part in the equation can be viewed as an external disturbance. By denoting the disturbance by D and substituting (6) for $f_j(t-d_i)$ in (21), we obtain the following closed-loop equation of the system:

$$\dot{q}_j(t)=D-\sum_{i\in Q_j}\left[C_P\{q_j(t-d_i)-q_j^T\}+C_I\int_0^{t-d_i}\{q_j(t)-q_j^T\}\,\mathrm{d}t\right]. \tag{22}$$

Now, we define the controller gains, C_P and C_I, to be

$$C_P=\frac{A}{|Q_j|},\quad C_I=\frac{B}{|Q_j|}, \tag{23}$$

where A and B are some positive constants. It is not difficult to see that the open-loop transfer function of the closed-loop system (22) is given by

$$F(s)=\left(\frac{A}{|Q_j|}\frac{1}{s}+\frac{B}{|Q_j|}\frac{1}{s^2}\right)\sum_{i\in Q_j}e^{-d_is}, \tag{24}$$

which is obviously a special case with $\rho_i=\frac{1}{|Q_j|},\ \forall i\in Q_j$ of

$$F(s)=\left(\frac{A}{s}+\frac{B}{s^2}\right)\sum_{i\in Q_j}\rho_ie^{-d_is}, \tag{25}$$

where $\rho_i\geqslant 0,\ \forall i\in Q_j$ and $\sum_{i\in Q_j}\rho_i\leqslant 1$. From now on, we use this generalized form of open-loop transfer function to find the stability condition.

First, we consider a single source case, i.e., $|Q_j|=1$ with $d_1=d$ and $\rho_1=1$. Note that this case is equivalent to the multiple source case with homogeneous delays $d_i = d$ and $\rho_i = \frac{1}{|Q_j|}$, $\forall i \in Q_j$. Then, the open-loop transfer function becomes

$$F(s) = \underbrace{\left(\frac{A}{s} + \frac{B}{s^2}\right)}_{\triangleq G(s)} e^{-ds}, \tag{26}$$

and letting $s = j\omega$ yields

$$F(j\omega) = \underbrace{\left(-\frac{B}{\omega^2} - j\frac{A}{\omega}\right)}_{\triangleq G(j\omega)} e^{-j\omega d}. \tag{27}$$

In [20], we already studied the asymptotic stability of the closed-loop system given by (22), so we only state two important theorems and one corollary. Their proofs can be found in [20].

Theorem 3. *The closed-loop system with a single source is asymptotically stable if and only if its delay d is bounded by*

$$0 \leqslant d < \frac{\arccos\left(\frac{B}{\bar{\omega}^2}\right)}{\bar{\omega}}, \tag{28}$$

where $\bar{\omega}$ is a unique $\omega > 0$ such that $|F(j\omega)| = 1$.

The above theorem shows the upper bound of the delay for a single source system to be asymptotically stable. It is, however, difficult to apply the stability condition (28) as it is to the design of a controller. We modify the condition into a usable form in the following corollary.

Corollary 1. *Let $\alpha = Ad$ and $\beta = Bd^2$. Then the closed-loop system with a single source is asymptotically stable if and only if*

$$0 < \alpha < \frac{\pi}{2} \quad \text{and } 0 < \beta < \omega_1^2 \cos\omega_1, \tag{29}$$

where ω_1 is the unique solution of $\alpha = \omega \sin\omega$ for $0 < \omega < \pi/2$.

Lastly, the stability condition for multiple sources of heterogeneous delays can be given by the theorem below.

Theorem 4. *The closed-loop system with multiple sources is asymptotically stable for all $0 \leqslant d_i \leqslant \bar{d}$ and for all ρ_i satisfying $\sum_{i \in Q_j} \rho_i \leqslant 1$ if and only if the closed-loop system of single source with delay $\bar{d}$ is asymptotically stable.*

Consequently, once the upper bound $\bar{d}$ of all the round-trip delays is known, the stable gain for the heterogeneous-delay case can be obtained from $A = \alpha/\bar{d}$ and $B = \beta/\bar{d}^2$ where α and β satisfy (29).

4.4. Optimal gain and estimation of $|Q_j|$

In [20], we already numerically showed that for given (α, β) satisfying (29), the asymptotic decay rate is maximized when $(\alpha, \beta) = (0.5, 0.1)$. Therefore, we use $(A, B) = (0.5/\bar{d}, 0.1/\bar{d}^2)$ as a stable and optimal controller gain. Based on this pair (A, B), we set the controller gain as $(C_P, C_I) = (A/|\widehat{Q}_j|, B/|\widehat{Q}_j|)$ where $|\widehat{Q}_j|$ is the estimate of $|Q_j|$. The key is that (i) By Theorem 4, $|Q_j|$ should be overestimated and (ii) $|\widehat{Q}_j|$ should be close to $|Q_j|$ because if it is overestimated too much, i.e., large $|\widehat{Q}_j|$, then the convergence speed will decrease severely. The details on this estimation can be found in [20].

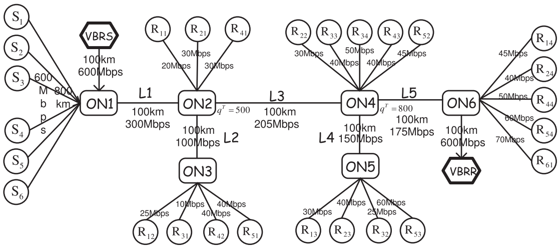


Fig. 4. Multiple-link configuration.

5. Simulation results

In this section, we verify through discrete-event simulations that the proposed algorithm works as designed and the proposed LB (Locality-Based) feedback consolidation algorithm results in the better transient performance than the hard-synchronization based consolidation algorithm [12][2], which we call WFA (Wait For All).

We examine the proposed algorithm in the multiple-link configuration shown in Fig. 4. There are 6 sessions and 21 VSs, and the VBR background traffic whose sender and receiver are respectively vbrs and vbrr. This VBR background traffic represents UDP traffic which shares the service overlay network. Multicast session i (1–5) has the sender S_i and its receivers R_{ij} (j = 1–4), and the unicast session 6 has one receiver R_{61}. The length of each link connecting a sender and the overlay node ON1 is different from each other, and the maximum length is set to be 800 km. The capacities of the links between senders and ON1 are equally set to 600 Mbps to ensure that no sessions are throttled there. To see the effectiveness of our feedback consolidation algorithm when some BCP arrivals are delayed significantly longer than the other BCP arrivals, the length of one receiver access link in each session is set to be 10,000 km while the other receiver access links are equally 50 km long.

The traffic model is summarized in Table 2, where we vary *MDR*, *PDR*, and arrival and departure times to see their impact on the network performance. Based on the network topology and traffic model, we can compute the theoretical fair rates over time as summarized in Table 3.

The simulation results without VBR background traffic are shown in Fig. 5. Each source's transmission rate in Fig. 5a exactly follows the theoretical fair rates given in Table 3 although there is a transient period whenever a session arrives or leaves. Fig. 5b shows that the queue lengths at L3 and L5 converge to its target value, 500 and 800 packets, in steady state. The reason for the empty queue at L3 in $[0, 1)$ and $[4, 5)$ is that no sessions are locally bottlenecked at L3 in those periods.These results provide an evidence that the local stability condition we found in

[2] In the WFA consolidation algorithm, each branching node must receive at least one BCP from all participating branches for the feedback consolidation operation.

Table 2
The traffic model.

Session	*MDR* (Mbps)	*PDR* (Mbps)	Arrival (s)	Departure (s)
1	15	150	1	3
2	20	150	2	4
3	10	20	0	∞
4	10	150	0	∞
5	25	150	0	∞
6	30	150	0	∞

Table 3
Theoretical fair rates (Mbps) over time (sec).

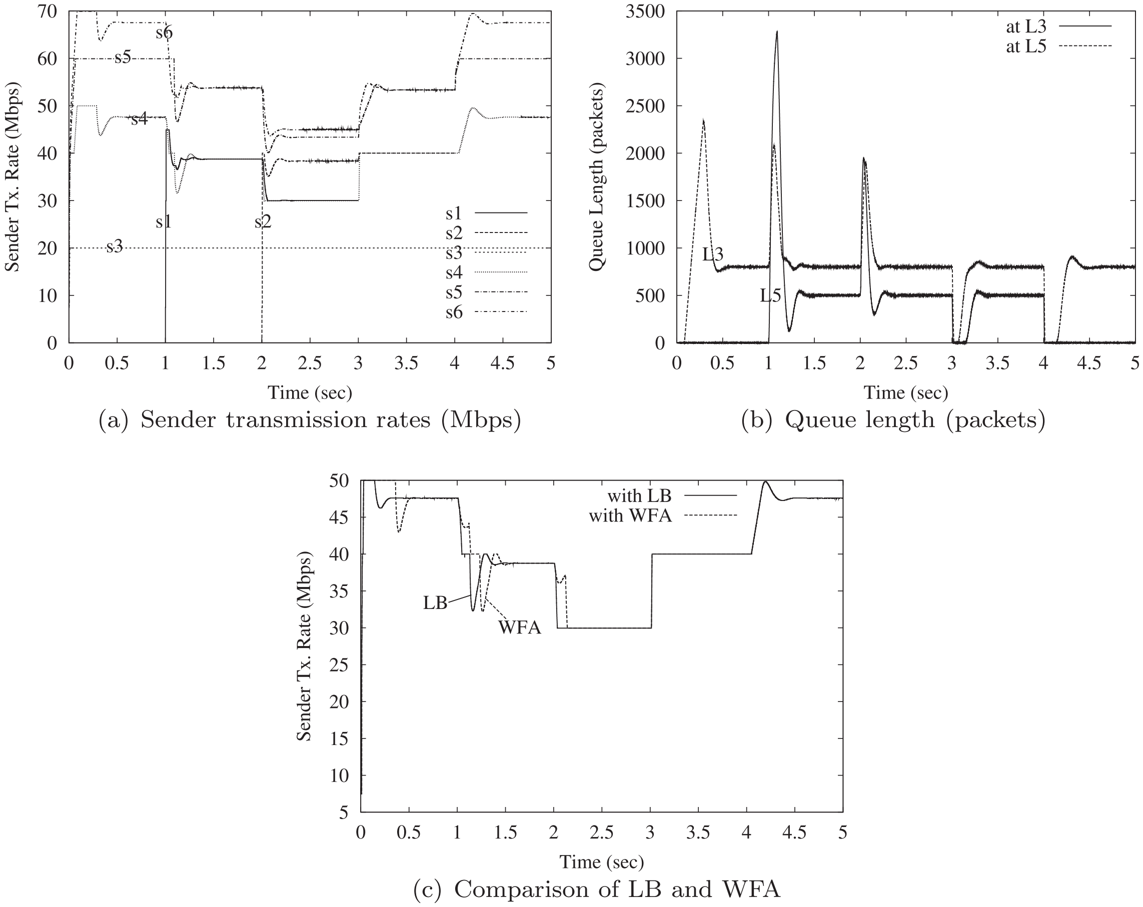
Session	0–1	1–2	2–3	3–4	4–5
1	–	38.75	30	–	–
2	–	–	38.3	40	–
3	20	20	20	20	20
4	47.5	38.75	30	40	47.5
5	60	53.75	43.3	53.3	60
6	67.5	53.75	45	53.3	67.5

Section 4 may serve as the global stability condition as well. Fig. 5c compares the performance of the two consolidation algorithms. Overall, LB yields better and more rapid transient performance than WFA as we expected in Section 2, which is because LB experiences smaller consolidation delay than WFA.

We also examine how the performance of the proposed algorithm is affected by the VBR background traffic, which is generated by superimposing 21 different H.26L encoded video clips [21] and has the average rate of approximately 60 Mbps as in Fig. 6a. The representative results are shown in Fig. 6b, c. Compared to the rate trace of S_5 in Fig. 5a, the one in Fig. 6b is shifted down approximately by 10 Mbps due to the addition of VBR traffic and includes high-frequency fluctuation. Lastly, the queue length shown in Fig. 6c fluctuates around its target value 800 packets. In brief, we can conclude that the unpredictable high-frequency traffic can lead to the high-frequency oscillation but never causes the system instability, which means that the performance is well bounded under our control.

6. Conclusions

In this paper, we proposed a distributed max–min flow control framework for multi-rate multicast flows focusing on the fair rate allocation and the feedback consolidation in service overlay network. The proposed fair rate allocation algorithm is highly scalable because it utilizes only the aggregate flow information for the rate computation. Our locality-based feedback consolidation algorithm reduces the consolidation delay and solves the feedback explosion problem by limiting the number of BCPs to that

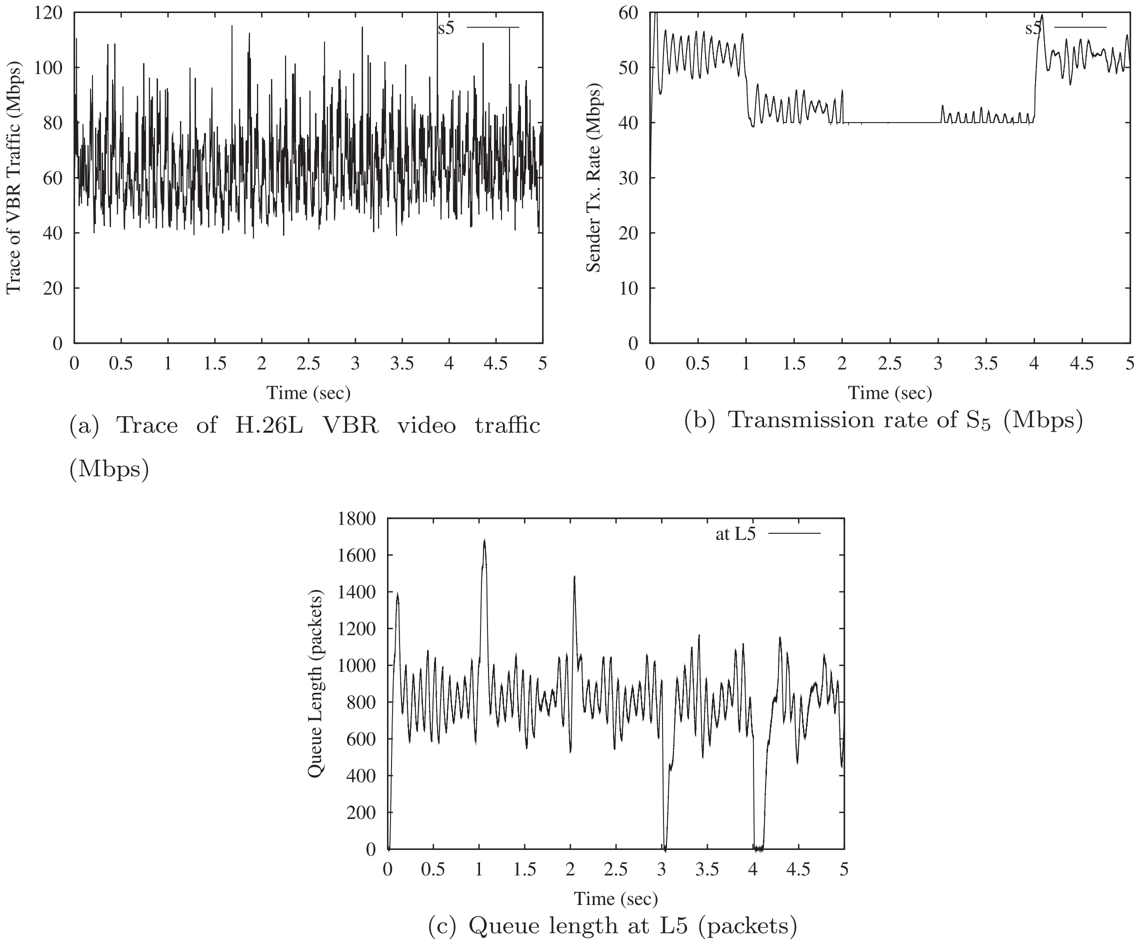


(a) Sender transmission rates (Mbps)

(b) Queue length (packets)

(c) Comparison of LB and WFA

Fig. 5. Results without VBR background traffic.



(a) Trace of H.26L VBR video traffic (Mbps)

(b) Transmission rate of S_5 (Mbps)

(c) Queue length at L5 (packets)

Fig. 6. Results with VBR background traffic.

of FCPs at every link. We mathematically showed that the proposed algorithm achieves the minimum plus max–min fairness and target queue length, and consequently the full link utilization at steady state. Moreover, we found the stability condition in a usable form taking into account heterogeneous round-trip delays. Simulation results verified that the proposed multi-rate multicast flow control scheme works as designed and the proposed feedback consolidation algorithm outperforms the existing hard-synchronization approach.

Acknowledgement

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-C1090-0902-0006).

References

[1] Z. Duan, Z.-L. Zhang, Y.T. Hou, Service overlay networks: SLAs, QoS, and bandwidth provisioning, IEEE/ACM Trans. Networking 11 (6) (2003) 870–883.

[2] J. Fan, M.H. Ammar, Dynamic topology configuration in service overlay networks: a study of reconfiguration policies, in: Proc. IEEE INFOCOM, Barcelona, Spain, 2006.

[3] N. Feamster, L. Gao, J. Rexford, How to lease the internet in your spare time, ACM SIGCOMM Comp. Commun. Rev. 37 (1) (2007) 61–64.

[4] Cabo: concurrent architectures are better than one. http://www.nets-find.net/Cabo.php.

[5] M. Bishop, S. Rao, K. Sripanidkulchai, Considering priority in overlay multicast protocols under heterogeneous environments, in: Proc. IEEE INFOCOM, Barcelona, Spain, 2006.

[6] Y. h. Chu, A. Ganjam, T.S.E. Ng, S.G. Rao, K. Sripanidkulchai, J. Zhan, H. Zhang, Early experience with an internet broadcast system based on overlay multicast, in: Proc. USENIX Annual Technical Conference 2004, Boston, MA, 2004.

[7] S. Sarkar, L. Tassiulas, Fair allocation of utilities in multirate, multicast networks: a framework for unifying diverse fairness objectives, IEEE Trans. Automatic Control 47 (6) (2002) 931–944.

[8] K. Kar, S. Sarkar, L. Tassiulas, A scalable low overhead rate control algorithm for multirate multicast sessions, IEEE J. Selected Areas Commun. 20 (8) (2002) 1541–1557.

[9] S. Sarkar, L. Tassiulas, Fair allocation of utilities in multirate multicast networks, in: Proc. 37th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, 1999.

[10] L. Roberts, Rate based algorithm for point to multipoint ABR services, in: ATM FORUM, 1994.

[11] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the h.264/avc standard, IEEE Trans. Circuits Sys. Video Tech. 17 (9) (2007) 1103–1120.

[12] S. Fahmy, R. Jain, R. Goyal, B. Vandalore, S. Kalyanaraman, Feedback consolidation algorithms for ABR point-to-multipoint connections, in: ATM Forum, 1997.

[13] K.J. Astrom, B. Wittenmark, Computer Controlled Systems: Theory and Design, Prentice-Hall, NJ, Englewood Cliffs, 1984.

[14] B.R. Barmish, New Tools for Robustness of Linear Systems, MacMillan, New York, 1994.

[15] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, New Jersey, 1992.

[16] C.F. Su, G. de Veciana, J. Walrand, Explicit rate flow control for ABR services in ATM networks, IEEE/ACM Trans. Networking 8 (3) (2000) 350–361.

[17] S. Chong, S.H. Lee, S.H. Kang, A simple, scalable, and stable explicit rate allocation algorithm for max–min flow control with minimum rate guarantee, IEEE/ACM Trans. Networking 9 (3) (2001) 322–335.

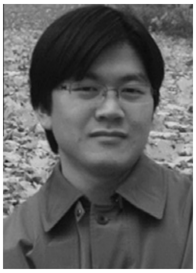
[18] L. Benmohamed, S.M. Meerkov, Feedback control of congestion in packet-switching networks: the case of multiple congested nodes, Int. J. Commun. Sys. 10 (5) (1997) 227–246.

[19] G.F. Franklin, J.D. Powell, M.L. Workman, Digital Control Systems, Addison-Wesley, 1990.

[20] J. Cho, S. Chong, Stabilized max–min flow control using PID and PII^2 controllers, IEICE Trans. Commun. E88-B (8) (2005) 3353–3364.

[21] Video traces for network performance evaluation. http://trace.eas.asu.edu/.

Hyang-Won Lee received his B.S., M.S. and Ph.D. degrees all in Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2001, 2003 and 2007, respectively. He is currently a Postdoctoral Research Associate at the Massachusetts Institute of Technology, Cambridge, MA. His research interests are in the areas of congestion control, wireless resource allocation, robust network design.



Jeong-woo Cho received his B.S., M.S., and Ph.D. degrees in Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2000, 2002, and 2005, respectively. From September 2005 to July 2007, he was with the Telecommunication R&D Center, Samsung Electronics, Suwon, Korea, as a Senior Engineer. From August 2007 to July 2008, he was a Senior Researcher in the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. From August 2008, he has been a Postdoc at the Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

His research interests include performance evaluation, mean field theory, network calculus, and cognitive radio networks.

Song Chong received the B.S. and M.S. degrees in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1995. Since March 2000, he has been with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, where he is a Professor and the Head of the Communications and Computing Group of the department. Prior to joining KAIST, he was with the Performance Analysis Department, AT&T Bell Laboratories, New Jersey, as a Member of Technical Staff. His current research interests include wireless networks, future Internet, and human mobility characterization and its applications to mobile networking. He has published more than 80 papers in international journals and conferences.

He is an Editor of Computer Communications and Journal of Communications and Networks. He has served on the Technical Program Committee of a number of leading international conferences including IEEE INFOCOM and ACM CoNEXT. He serves on the Steering Committee of WiOpt and was the General Chair of WiOpt '09. He is currently the Chair of Wireless Working Group of the Future Internet Forum and the Vice President of Information and Communication Society of Korea.