IEICE TRANS. COMMUN., VOL.E85–B, NO.3 MARCH 2002

PAPER

Active Multicast Congestion Control with Hop-by-hop Credit-based Mechanism

Jong-Kwon LEE[†] *and* **Tag Gon KIM**[†], *Nonmembers*

SUMMARY This paper proposes a credit-based congestion control scheme for multicast communication which employs application-specific processing at intermediate network nodes. The control scheme was designed not only to take advantage of credit-based flow control for unicast communication, but also to achieve flexibility supported by active network technology. The resultant active multicast congestion control scheme is able to meet the different requirements of various multicast applications in terms of reliability and end-to-end latency. The performance of the proposed control scheme was evaluated using both discrete-event simulations and experiments on a prototype active network implementation. The results show that the proposed scheme performs very well in terms of fairness, responsiveness, and scalability. The implementation experiences also confirmed the feasibility of the scheme in practice.

key words: *multicast, congestion control, credit-based flow control, active networking*

1. Introduction

In spite of its efficient data distribution, multicast communication is by definition greedier in bandwidth than unicast communication. Furthermore, multicast flows insensitive to congestion are likely to cause simultaneous congestion collapse in many parts of the network. Therefore, multicast congestion control is becoming important and essential for the deployment of multicast services [1], [2].

A major problem in multicast congestion control is the scalability of the feedback mechanism. The scalability problem arises when the feedback information about network congestion or receivers' status should be provided to the multicast source. Any feedback packet from receivers should not be simply neglected as in loss recovery protocols. Instead, the feedback information must be summarized at intermediate nodes or receivers. As a result, it is necessary to design a feedback consolidation mechanism rather than just a feedback suppression mechanism.

A fairness issue is another important problem in multicast congestion control. The fairness problem has been classified into two categories: among different flows (*inter-fairness*) and among the same multicast group receivers (*intra-fairness* or *inter-receiver fairness*) [3]. Fairness among different flows is not subject to only a multicast congestion control problem. It has been also considered as an important issue in the unicast communication on the Internet. The problem can be transformed into a question of whether a multicast traffic flow is TCP-friendly or not. A TCP-friendly congestion control scheme consumes no more bandwidth than a conforming TCP connection running under comparable conditions. The fairness problem among the same group receivers is a question of who the source transmission rate is adapted to. If the source rate is determined by the slowest receiver, fast receivers may be unhappy. On the other hand, if some slow receivers are neglected, the links to these receivers may be overloaded. Choosing an appropriate fairness criterion is a policy issue and may be varied according to the relevant multicast applications.

Several schemes have been proposed for congestion control of real-time and reliable multicast applications [3]–[6]. Most of them were end-to-end mechanisms implemented at the transport layer: only the end hosts, not the intermediate network nodes, are involved in congestion control. However, there has been no scheme suited for every multicast application. As multicast applications have a diverse spectrum in terms of reliability semantics and end-to-end latency requirements [7], no single control strategy may be applied for all the applications. For the scalability of the feedback mechanism, some used NACK packets as congestion signals, and others used tree structure for hierarchical feedback of ACK packets. With respect to the inter-receiver fairness, most reliable multicast congestion control schemes adopted a fairness policy such as *"a source should adjust its transmission rate to the most bottleneck receiver among the group members"*, called *worst-path fairness*.

Hop-by-hop approaches to multicast congestion control have not been preferred, for they are less flexible than end-to-end approaches. While an end-to-end congestion control scheme can be varied partly in its control algorithm according to particular applications, a hop-by-hop scheme does not have such flexibility. Its control algorithm must be implemented inside routers in advance, thus being difficult to modify according to the applications. However, feedback aggregation at intermediate nodes is advantageous since it eliminates the overhead associated with organizing receivers in the end-to-end feedback mechanism.

This paper proposes a hop-by-hop multicast con-

Manuscript received April 18, 2001.
Manuscript revised September 3, 2001.

[†]The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea.

gestion control scheme which can achieve as much flexibility as end-to-end approaches. As a hop-by-hop approach to multicast congestion control, we adopted a credit-based mechanism. The credit-based flow control was first proposed for unicast communication over ATM networks [8] and later extended to the network layer (or IP layer) mechanism for deployment on the Internet [9]. In [10], a credit-based mechanism was applied to congestion feedback for the multicast of adaptively encoded video. Based on these previous researches, we developed a credit-based multicast flow control scheme supported by active networking technology. Active networking has recently emerged as a new networking paradigm, in which network elements are programmable and can perform application-specific processing on user data [11], [12]. Therefore, the proposed scheme allows us to easily trade off data reliability against latency requirements by executing corresponding programs at the intermediate active nodes. In fact, the scheme may be better suited for reliable multicast applications because the credit-based mechanism can support reliable data delivery without the help of transport-layer protocol. However, with active networking support, it can provide a general framework of multicast congestion control for both real-time and reliable multicast applications.

The rest of this paper is organized as follows. Section 2 describes the details of the credit-based multicast flow control algorithm. In Sect. 3, an active networking method to enhance the flexibility of the proposed credit-based scheme is explained. Sections 4 and 5 show the results of performance evaluation of the proposed scheme using discrete-event simulations and experiments on a prototype active network implementation. Finally, we conclude this paper in Sect. 6.

2. Credit-based Multicast Flow Control

Before describing the details of the proposed scheme, we briefly review credit-based flow control for unicast communication [8]. Between two adjacent nodes on the communication path of a packet flow, a credit packet is sent from a downstream node to an upstream node whenever the downstream node has forwarded a data packet. Credits reflect the amount of buffer space available at the downstream node and give the upstream node permission to transmit data packets. Whenever a node transmits a packet, it consumes one credit and transfers a credit to its upstream node. If a node has no credits, it must wait for one or more credits to arrive before transmitting a packet. For efficiency, several credits are collected into a single credit packet at each node before being transmitted to the upstream node.

A credit-based flow control protocol is a hop-by-hop, window-based mechanism. Hop-by-hop flow control mechanisms have been generally implemented at the link layer in a homogeneous network. However, the network today is composed of interconnected subnetworks, which are usually covered by different link layer protocols. For this reason, the proposed flow control mechanism is implemented at the network layer or IP layer of the Internet as in [9]. Implementation at the network layer is particularly advantageous in case of multicast congestion control. This is because the layer is on the same level as that of multicast distribution tree.

In developing a credit-based multicast flow control mechanism, two assumptions are made: dynamic flow-setup and static buffer allocation. The flow-setup is done dynamically by the first packet of that flow. With active networking, the flow-setup process may be programmed into packets. The static buffer allocation was assumed for simplicity. Although dynamic buffer allocation is expected to be advantageous in the utilization of buffer space, the buffer allocation problem is beyond the scope of this paper. From the viewpoint of deployment, it was not assumed that all the routers should be active and participate in the credit-based multicast flow control. Consequently, the proposed scheme can be applied gradually to the current Internet.

2.1 Multicast Credit Update Protocol

Table 1 shows node variables used in the credit update protocol for multicast communication. Hosts and intermediate nodes which are involved in credit-based multicast flow control should maintain the variables of their own. We call these nodes *flow-control nodes* and the rest in the same multicast tree *non-flow-control nodes*. Besides, we let a variable, e.g. D_i, maintained at the flow-control node N be denoted by D_i^N.

Note that a credit balance ($CB_{i,t}$) is maintained for each of the next downstream flow-control nodes (T_i) and the other variables ($B_{i,d}$, $FC_{i,d}$, and $TC_{i,d}$) are for each of the next downstream multicast nodes

Table 1 Notations for node variables in credit update protocol for multicast flow i.

D_i	A set of IDs of the next downstream multicast nodes to which copied data packets of flow i are forwarded by the current node. This is determined when multicast tree is constructed.
T_i	A set of tags of the next downstream flow-control nodes which send credit packets for flow i to the current node. Each tag is attached to the credit packets.
$B_{i,d}$	Buffer allocation for $d \in D_i$.
$FC_{i,d}$	Forwarding sequence counter for $d \in D_i$.
$TC_{i,d}$	Transmission sequence counter for $d \in D_i$.
$CB_{i,t}$	Credit balance for $t \in T_i$.
B_i	Effective buffer allocation for updating credit balance of the upstream flow-control node.
FC_i	Effective forwarding sequence counter for updating credit balance of the upstream flow-control node.
$\delta_i(\cdot)$	A mapping from T_i to D_i.

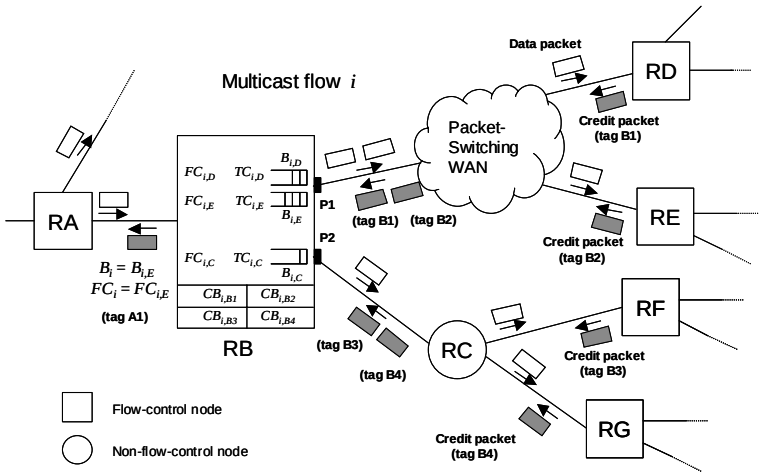


Fig. 1 Example of credit-based multicast flow control.

(D_i). Credit balances are updated by credit packets coming from downstream branches. If one of the next downstream nodes (D_i) is a non-flow-control node, credit packets from that downstream branch are those which have originated from the further downstream flow-control nodes (T_i). Furthermore, when a node X maintains a credit balance for one of the next downstream flow-control nodes, e.g. $Z \in T_i^X$, increasing and/or decreasing the credit balance is closely related to the transmission sequence counter for the next downstream node $Y \in D_i^X$ on the branch from X to Z. Therefore, a mapping from T_i to D_i is required.

Now, let us define a mapping from T_i to D_i, $\delta_i(\cdot) : T_i \rightarrow D_i$, as follows: if we denote the current node by N, $\delta_i^N(\cdot)$ maps each tag of downstream flow-control nodes, $t \in T_i^N$, to a node $d \in D_i^N$ which is on the path from N to the node with tag t. For example, in Fig. 1, multicast router C is a non-flow-control node while multicast routers F and G are flow-control node with tags B3 and B4, respectively. Therefore, $C \in D_i^B$, $B3 \in T_i^B$, and $B4 \in T_i^B$. Since router C is on the path from router B to router F, a mapping from tag B3 to router C, i.e. $\delta_i^B(B3) = C$, is established at router B. In case of router G, $\delta_i^B(B4) = C$ is also set up for the same reason.

With these node variables, the multicast credit update protocol for flow i is described as follows.

- **Operation of non-flow-control nodes**
  Non-flow-control nodes simply forward data/credit packets to the next downstream/upstream nodes.

- **Data packet forwarding**
  A flow-control node U forwards data packets to the next downstream multicast node $d \in D_i^U$ only if every $CB_{i,t}^U$ such that $\delta_i^U(t) = d$ is positive. When a packet is forwarded to node d, all the corresponding $CB_{i,t}^U$ decreases by one and both $FC_{i,d}^U$ and $TC_{i,d}^U$ increase by one.

- **Deadlock recovery from data packet loss**
  Forwarding sequence counters (FC's) at the downstream flow-control nodes are synchronized with the value of transmission sequence counters (TC's) at the upstream flow-control nodes by carrying the latter within data packets. This is necessary for deadlock recovery in case of data packet loss [9].

- **Feedback credit packets**
  Each flow-control node sends a credit packet backward to its upstream node whenever a certain condition, called *feedback condition*, is satisfied. The feedback condition is dependent on the inter-receiver fairness criterion used in the current application. For example, we can choose a feedback condition such as *"at least M data packets have been forwarded to each of the next downstream multicast node."* In this condition, M is called *credit update unit*.

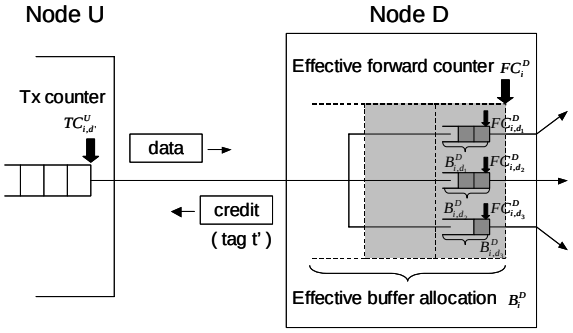
- **Credit balance update**
  When an upstream flow-control node U receives credit packets from the next downstream flow-control node D with tag $t' \in T_i^U$, the credit balance $CB_{i,t'}^U$ of the upstream node U is updated as

$$\begin{aligned} CB_{i,t'}^U &= B_i^D - (TC_{i,d'}^U - FC_i^D) \\ &= B_i^D + FC_i^D - TC_{i,d'}^U \end{aligned} \quad (1)$$

  where

$$d' = \delta_i^U(t') \in D_i^U. \quad (2)$$

  B_i^D and FC_i^D are the effective values of buffer allocation and forwarding sequence counters, respectively, at node D (see Fig. 2). These values are required to represent the buffer space limit of the downstream node D as if the downstream node had a single buffer and a single forward sequence



Credit balance = Buffer Allocation – (Forward counter – Tx counter)

$$CB^{U}_{i,t'} = B^{D}_{i} - (TC^{U}_{i,d'} - FC^{D}_{i}) = B^{D}_{i} + FC^{D}_{i} - TC^{U}_{i,d'}$$

Fig. 2 Effective buffer allocation and transmission sequence counter to update credit balance.

counter. Thus the quantity, $TC^U_{i,d'} - FC^D_i$, represents the outstanding credits which are data packets of multicast flow i that the upstream node U has transmitted but the downstream node D has not regarded as forwarded. Therefore, the updated credit balance means the number of packets that are currently allowed to be received by the downstream node.

- **Determining feedback values**
  From another point of view, $B^D_i + FC^D_i$ in Eq. (1) can be interpreted as the largest sequence counter of the allowed packets to be received by the downstream node D without further credit updates. B^D_i and FC^D_i are determined based on the inter-receiver fairness criterion when they are carried within the credit packet. For example, if we choose *worst-path fairness*, they are set in the downstream node D as

$$B^D_i = B^D_{i,k}, \tag{3}$$

$$FC^D_i = FC^D_{i,k}, \tag{4}$$

where k is

$$k = \arg\min_{d \in D_i} \{B^D_{i,d} + FC^D_{i,d}\}. \tag{5}$$

The above protocol can be better understood with the following example.

2.2 Illustrating Example

Figure 1 shows an example of the credit-based multicast flow control. It was assumed that data delivery should be reliable and thus traffic rate should be adjusted to the most bottleneck path. Credit update unit M was assumed to be 2.

Note that router B must make three copies of the data packet received from router A, and address them to C, D, and E, respectively. Two copies of the packet are forwarded through interface $P1$ since the network among multicast routers B, D, and E is a packet-switching WAN, which is incapable of multicasting. Also note that credit packets from routers D and E are received at the same interface $P1$ of router B. The credit packets from different downstream nodes are identified by their own tags (B1 and B2).

On the other hand, C is a multicast router, but not a flow-control node. Router B forwards data packets to router C and receives credit packets from routers F and G (with tags B3 and B4, respectively), all through interface $P2$. Therefore, in router B, $D_i = \{C, D, E\}$ and $T_i = \{B1, B2, B3, B4\}$. Then, $\delta^B_i(\cdot)$ is defined as:

$$\delta^B_i(B1) = D, \quad \delta^B_i(B2) = E,$$

$$\delta^B_i(B3) = C, \quad \delta^B_i(B4) = C.$$

Whenever receiving credit packets, router B updates credit balances by Eq. (1) as follows:

$$CB^B_{i,B1} = B^D_i + FC^D_i - TC^B_{i,D}, \tag{6}$$

$$CB^B_{i,B2} = B^E_i + FC^E_i - TC^B_{i,E}, \tag{7}$$

$$CB^B_{i,B3} = B^F_i + FC^F_i - TC^B_{i,C}. \tag{8}$$

$$CB^B_{i,B4} = B^G_i + FC^G_i - TC^B_{i,C}. \tag{9}$$

Data packets can be forwarded to router C only if both $CB^B_{i,B3}$ and $CB^B_{i,B4}$ are positive.

At the moment in this figure, the path to router E seems to be a bottleneck since the buffer for E has the least amount of available capacity. The incoming traffic rate to router B must be adjusted to the service rate of this buffer. Therefore, if at least two new packets have been forwarded, a new credit packet is created and transmitted backwardly to router A. Because the value $B^B_{i,d} + FC^B_{i,d}$ is minimum for $d = E$, the values of B^B_i and FC^B_i in the credit packet are determined by Eq. (3), (4), and (5):

$$B^B_i = B^B_{i,E}, \tag{10}$$

$$FC^B_i = FC^B_{i,E}. \tag{11}$$

With this credit packet, router A updates the credit balance for router B as

$$CB^A_{i,A1} = B^B_i + FC^B_i - TC^A_{i,B}. \tag{12}$$

2.3 Pros and Cons of Credit-based Approach to Multicast Congestion Control

A credit-based flow control inherently has several advantages. First of all, it can lead to no packet loss if buffers for each of the downstream multicast nodes

are allocated sufficiently. Other advantages include shorter feedback delay compared with end-to-end control schemes and the provision of fairness among many flows by per-flow queuing when combined with a fair scheduling algorithm. When the credit-based mechanism is applied to multicast congestion control, additional benefits can be obtained: the feedback aggregation at intermediate nodes occurs in a scalable fashion. Moreover, such a hop-by-hop scheme can operate independent of the transport-layer loss recovery protocol for reliable multicast services.

On the other hand, the hop-by-hop credit-based scheme has some problems in flexibility for use in multicast applications. The scheme must be implemented at the routers, and the parameters and control strategies configured in advance of operation. However, because multicast applications have different requirements in reliability and end-to-end latency, a pre-configured protocol may not be suitable for use in certain multicast applications. The scheme also requires more buffers and processing power in flow-control nodes since each flow-control node should maintain per-flow states and perform credit update protocol.

In order to solve these problems of credit-based multicast flow control, we developed a programmable mechanism which is supported by active networking technology. This is explained in the next section.

3. Enhancing Flexibility with Active Networking Support

Active networking has recently appeared as a new networking paradigm to increase the flexibility and utilization of the existing networks [11], [12]. Active networking allows network elements to perform customized computations on user data. Active services could be deployed as application-level programs inside active routers or, equivalently, on servers collocated with routers in the current Internet. With active networking support, we can obtain application-specific services for various multicast applications within the network. In other words, active networking can be a promising solution for those problems described in Sect. 2.3. For example, we can easily modify the inter-receiver fairness criteria or trade off data reliability against latency requirements by executing corresponding programs at the nodes. It is not necessary to implement the flow control scheme statically, but only to load appropriate program codes dynamically carried within active packets. Consequently, a unified framework of congestion control for both real-time and reliable multicast applications can be obtained. In addition, memory requirements and processing power to perform the credit update protocol are provided by execution environments in the active nodes.

In the active networking environment, data and credit packets of the proposed flow control are active packets (or capsules) that contain program codes for the credit update protocol. The programs in the packets are loaded on the fly into the active nodes in the multicast tree, and then executed during the multicast session.

Notation

ThisNode.tag	Variable for the tag of current node to identify itself
CRED.B	Field in the CRED header to report buffer size
CRED.FC	Field in the CRED header to report forwarding counter
CRED.t	Field in the CRED header to report the tag of current node

```
1   Active data packet DATA
2     when DATA is arrived
3       if (DATA is the first packet of multicast flow i)
4         initialize node variables
5       end if
6       for each d ∈ D_i
7         if (buffer for d has available spaces)
8           copy DATA into buffer for d
9         end if
10      end for
11    end when
12    when buffer for d is scheduled
13      if (CB_{i,t} > 0 for all t s.t. δ_i(t) = d)
14        forward DATA toward downstream node d
15        TC_{i,d} ← TC_{i,d} + 1
16        FC_{i,d} ← FC_{i,d} + 1
17        CB_{i,t} ← CB_{i,t} − 1
18        if (FC_{i,d} = min_{d∈D_i}{FC_{i,d}} and FC_{i,d} mod M = 0)
19          find k = arg min_{d∈D_i}{B_{i,d} + FC_{i,d}}
20          create a credit packet CRED
21          CRED.B ← B_{i,k}
22          CRED.FC ← FC_{i,k}
23          CRED.t ← ThisNode.tag
24          forward CRED toward upstream node
25        end if
26      end if
27    end when
28  end DATA
```

```
1   Active credit packet CRED
2     when CRED is arrived
3       CB_{i,CRED.t} ← CRED.B + CRED.FC − TC_{i,δ_i(CRED.t)}
4     end when
5   end CRED
```

Fig. 3 Algorithms for active packet DATA and CRED.

For example, if the proposed active multicast congestion control scheme is employed by a reliable multicast application, the feedback condition for credit packets is determined as in Eq. (3), (4), and (5). Assuming that the credit update unit is expressed by M, the algorithms for the proposed credit-based multicast flow control implemented in the active packets are as shown in Fig. 3. Variables and functions in the algorithms which are not defined in the notation list are the same as in Sect. 2. The feedback condition for credit packets is expressed in the lines 18 to 25. The expression and procedure in these lines may be modified when different applications use different criteria for the inter-receiver fairness.

With this active mechanism, the proposed scheme can achieve flexibility in that it can satisfy the requirements of different multicast applications.

4. Performance Evaluation by Simulation

The performance of the proposed scheme was evaluated using both discrete-event simulations and experiments on a prototype implementation. Simulations were used to evaluate the performance of the flow control algorithm aside from implementation issues. However, the objective of experiments was mainly to investigate the feasibility of the proposed scheme instead of intensive performance evaluation.

Discrete-event simulations were carried out by using the DEVSim++ simulation environment [13], which is a C++ realization of the DEVS (Discrete Event Systems Specification) formalism [14]. We focused on the reliable multicast, though the proposed scheme can be applied to other classes of multicast applications. Consequently, the worst-path fairness was chosen as the inter-receiver fairness criterion.

The following performance indices were used in the simulation.

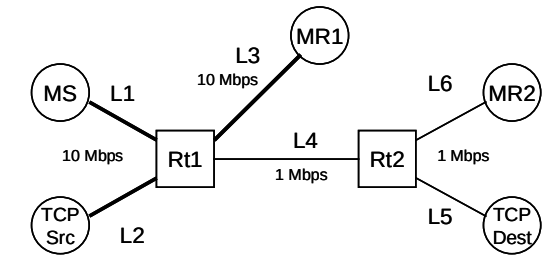
- **Fairness**
  The inter-fairness, i.e. fairness among different flows, was used to evaluate the proposed scheme. Especially, it was investigated if the proposed scheme is TCP-friendly, i.e., how fairly the multicast flow controlled by the proposed scheme shares the bandwidth with a competing TCP flow.

- **Responsiveness**
  In order to be effective, feedback-based congestion control schemes must react in a timely fashion to changes in the network's congestion status.

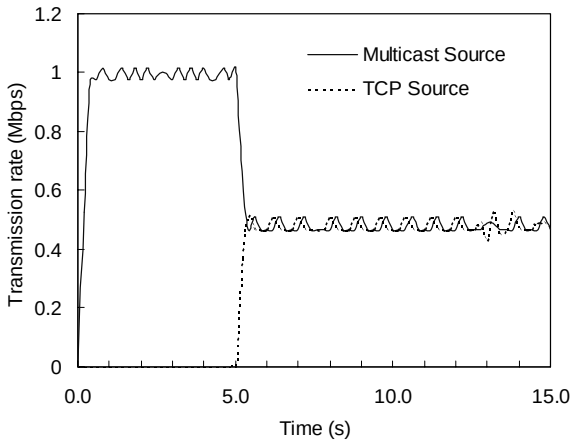
- **Scalability**
  Scalability is another important performance measure of a multicast congestion control scheme. The degree of throughput degradation as the group size increases was used to evaluate the scalability of the proposed scheme.

In the simulation study, RMTP [16] was used as a transport-layer loss recovery protocol for reliable multicast services. The feedback condition for credit packets was chosen as the forwarding of at least 2 data packets to each of the next downstream node (i.e., credit update unit $M = 2$). The effect of credit update unit on the performance is dealt with in next section because it can be better investigated by experiments on the prototype implementation.

Figure 4 shows the fairness and responsiveness of the proposed scheme. The topology in Fig. 4(a) is similar to one of the reference topologies suggested in [15] for evaluating reliable multicast congestion control schemes. Such a simple multicast topology is sufficient for evaluating fairness and responsiveness of the proposed scheme.



(a) Topology for evaluating fairness and responsiveness.



(b) Source transmission rate: proposed scheme vs. TCP.

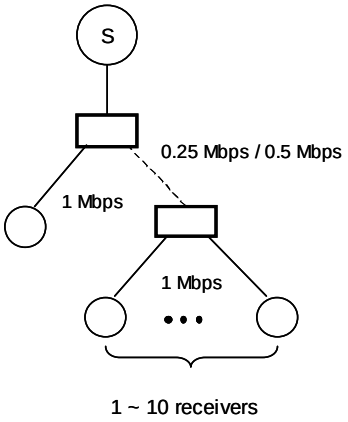
Fig. 4 Simulation results: fairness and responsiveness of the proposed scheme.

Packet sizes were assumed to be 512 bytes for data packets and 80 bytes for credit packets, respectively, and buffer allocation for each of the next downstream node in the routers was set at 5 packets. Each communication link had a propagation delay of 5 ms, and L1, L2, and L3 had bandwidths of 10 Mbps while the others 1 Mbps.

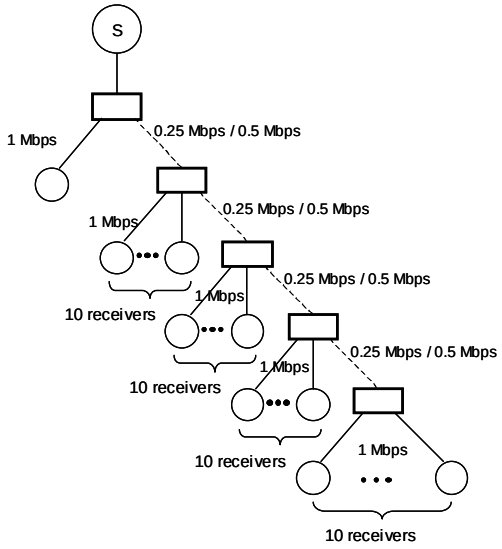
Figure 4(b) shows the transmission rates of the multicast source and the TCP source when the multicast session started at time 0 and the TCP session started at time 5 sec. At first, only multicast traffic used the bandwidth of bottleneck link L4. When unicast traffic crossing link L4 occurred, the multicast source rapidly reduced its transmission rate and eventually settled at a fair share of the bottleneck bandwidth with the unicast source. This simulation result shows that the multicast traffic controlled by the proposed flow control is responsive to bandwidth changes and shares the bottleneck bandwidth fairly with the unicast traffic in the steady state.

The scalability of the proposed scheme is shown in Fig. 5. All the parameters except link bandwidths were set at the same values as in the above simulation. Each link between hosts and routers had bandwidths of 1 Mbps, while the bandwidths of links between routers were set at the lower values—0.25 Mbps and 0.5 Mbps.

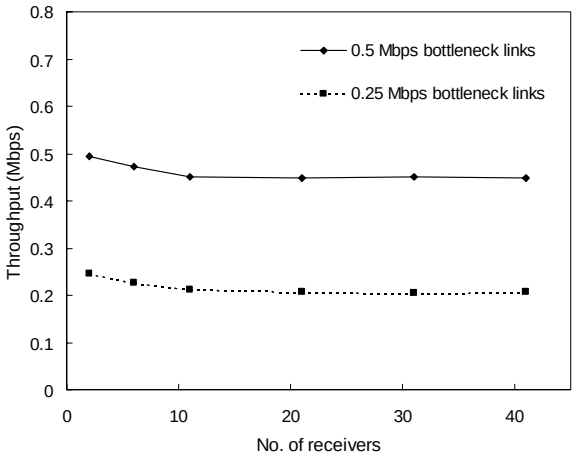
Firstly, the number of receivers was increased in the two-level multicast tree topology of Fig. 5(a). The



(a) Two-level multicast tree.



(b) Multi-level multicast tree.



(c) Average throughput vs. number of receivers

Fig. 5 Simulation results: scalability of the proposed scheme.

number of receivers connected to the second-level router was varied from 1 to 10. Then, the multicast group size was varied by increasing the level of multicast tree in which there were 10 receivers at each tree level

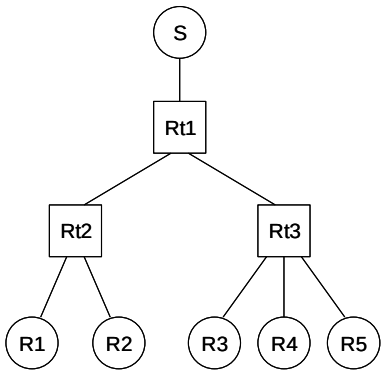


Fig. 6 Experimental multicast topology in prototype implementation.

(Fig. 5(b)). Average throughput of a multicast session was obtained from the measurements for 10 seconds in the steady state.

Figure 5(c) shows the result for the scalability of the proposed scheme. As the number of receivers connected to the same router was increased, the average throughput was degraded by about 10 %. Note that, however, it was maintained almost at the same value if the group size was varied by increasing the level of multicast tree, i.e., the number of routers. Even when there are many more receivers in the multicast group, the number of receivers or routers connected directly to a router will be still limited. Therefore, it is expected that the proposed scheme would be able to maintain its average throughput even with hundreds or thousands of receivers. The result shows that the proposed scheme is scalable with respect to the multicast group size.

5. Experiments with Prototype Active Network Implementation

To examine the feasibility of the proposed flow control with active networking support, we implemented a prototype of the proposed scheme using the Active Network Transport System (ANTS), which is an active networking execution environment implemented in Java [17]. The prototype implementation was run on Pentium II PCs under Windows 2000 Operating System. We emulated a multicast environment by creating multiple ANTS nodes on different PCs in a 155 Mbps ATM LAN. The ANTS nodes communicate with one another through UDP.

Experiments on the prototype of the proposed scheme were conducted using the topology shown in Fig. 6. The feedback condition was set to the same as in the simulation study except that the credit update unit M was varied to investigate its effect on performance. The link from Rt2 to R1 was a bottleneck with a bandwidth of 12.5 packets/s, and other links had bandwidths of more than 20 packets/s. The propagation delay of each link was assumed to be negligible.

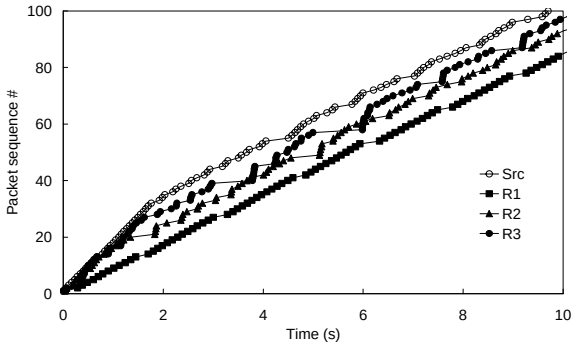


Fig. 7 Traces of source and receivers (buffer allocation = 5 packets, credit update unit = 1 packet).

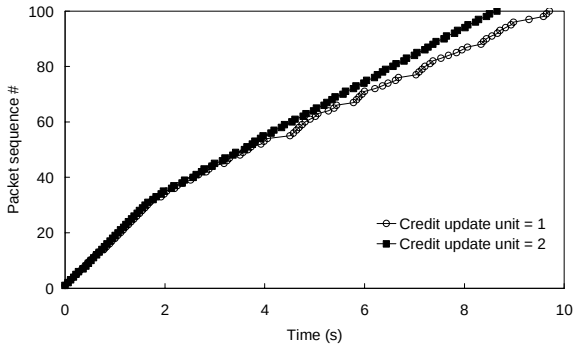


Fig. 9 Source behaviors for different credit update unit (buffer allocation = 5 packets).

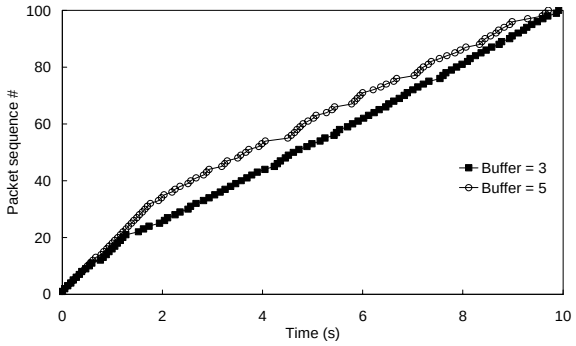


Fig. 8 Source behaviors for different buffer allocation (credit update unit = 1 packet).

Initially the source transmitted data packets at a rate of 20 packets/s, and then adjusted its transmission rate according to the feedback signals. The behavior of the source S and the receivers R1, R2, and R3 was observed during the communication.

Figure 7 shows the traces of the source and the receivers. The slope in this graph means the transmission or receiving rate of each host. It was observed that the receiving rate of R1 was slightly less than the bottleneck link rate of 12.5 packets/s from the beginning. The rates of the other receivers and the source were reduced from about 20 packets/s to the bottleneck rate in turn before 2 seconds elapsed. This is because the intermediate nodes regulated the traffic flows to their downstream nodes before the effect of the bottleneck receiver R1 propagated back to the source. This result confirms that the proposed scheme is able to respond quickly to changes in the network congestion level in the vicinities of the congested region.

Figures 8 and 9 show the results for different buffer allocation and credit update unit, respectively. In Fig. 8, it was observed that the source with buffer allocation of 3 packets reduces its rate earlier than the source with 5 packets. That is to say, the source with smaller buffer allocation adjusted its transmission rate more quickly. Large buffer allocation may be advantageous when there are sudden congestion occurrences in the network. With large buffer allocation, however, the effect of the local congestion propagates back to the source in a long delay. In addition, the larger is the buffer allocation, the longer is the average end-to-end delay in data delivery. This implies that the buffer allocation must be designed to minimize packet losses with as small capacity as possible. Figure 9 shows that a higher throughput is obtained with less frequent credit updates—about 10 packets/s vs. 8.5 packets/s. This result is straightforward because there is less processing and communication overhead with infrequent credit packets. If the buffer allocation becomes large, the proposed scheme can also operate with the larger values of credit update unit. However, it is considered that the experiments with such small values of credit update unit are enough to show the influence of credit update unit on the performance.

From these results, it is expected that an optimal throughput and delay performance can be obtained by choosing the appropriate values of buffer allocation and credit update unit. The experiments on the prototype implementation also showed that the proposed scheme with active networking support is feasible in practice.

6. Conclusions

This paper proposed an active multicast congestion control scheme with a hop-by-hop credit-based mechanism. The credit-based multicast flow control scheme includes several advantages such as no packet loss, short feedback delay, scalable feedback aggregation, and so on. Its weakness in flexibility can be complemented by adopting active network technology that supports application-specific processing at intermediate network nodes. Consequently, the proposed scheme can be applied to congestion control for both real-time and reliable multicast applications.

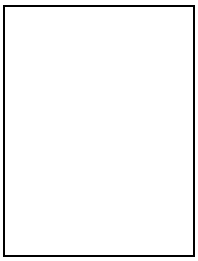
The proposed control mechanism was implemented in a prototype active network environment and its performance was evaluated using both discrete-event simulations and experiments. The simulation results showed

that the proposed scheme performed very well in terms of fairness, responsiveness, and scalability. The experimental results showed the effects of buffer allocation and credit update unit on performance, and also confirmed the feasibility of the proposed scheme in practice.

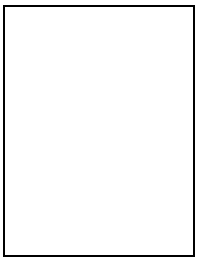
There are two problems to be considered as further research. One is the buffer allocation problem for the proposed credit-based multicast flow control scheme. The design of buffer allocation algorithm is important from the viewpoint of both cost and performance. The other is the study on the interactions between the proposed hop-by-hop scheme and the transport layer protocols for reliable multicast congestion control. It is expected that the combination of hop-by-hop and end-to-end mechanisms can bring positive effects on performance and flexibility in certain multicast applications.

References

[1] K. Obraczka, "Multicast transport protocols: a survey and taxonomy," IEEE Comm. Mag., vol.36, no.1, pp.94-102, Jan. 1998.
[2] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: a survey of protocols, functions, and mechanisms," IEEE J. Select. Areas. Comm., vol.15, no.3, pp.277-290, April 1997.
[3] I. Rhee, N. Balaguru, and G. N. Rouskas, "MTCP: scalable TCP-like congestion control for reliable multicast," Proc. IEEE INFOCOM'99, New York, NY, pp.1265-1273, March 1999.
[4] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," Proc. ACM SIGCOMM'96, Stanford, CA, pp.117-130, Aug. 1996.
[5] D. DeLucia and K. Obraczka, "Congestion control performance of a reliable multicast protocol," Proc. Int. Conf. Network Protocols '98, pp.168-176, 1998.
[6] S. J. Golestani and K. K. Sabnani, "Fundamental observations on multicast congestion control in the Internet," Proc. INFOCOM'99, New York, NY, PP.990-1000, March 1999.
[7] S. Paul, Multicasting on The Internet and Its Applications, Kluwer Academic Publishers, Norwell, MA, 1998.
[8] H. T. Kung, T. Blackwell, and A. Chapman, "Credit-based flow control for ATM networks: credit update protocol, adaptive credit allocation, and statistical multiplexing," Proc. ACM SIGCOMM, London, U.K., pp.101-114, Aug. 1994.
[9] K. Chang, IP Layer Per-Flow Queueing and Credit Flow Control, Ph.D Thesis, Harvard University, Jan. 1998.
[10] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive multicast of multi-layered video: rate-based and credit-based approaches," Proc. INFOCOM'98, San Francisco, CA, pp.1073-1083, March 1998.
[11] S. Ortiz Jr., "Active networks: the programmable pipeline," Computer, vol.31, no.8, pp.19-21, Aug. 1998.
[12] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," IEEE Comm. Mag., vol.35, no.1, pp.80-86, Jan. 1997.
[13] T. G. Kim, DEVSim++ User's Manual: C++ Based Simulation with Hierarchical, Modular DEVS Models, Systems Modeling Simulation Lab., KAIST, Taejon, Korea, 1994; ftp://sim.kaist.ac.kr/pub/DEVSim++-1.0/
[14] B. P. Zeigler, Multifacetted Modeling and Discrete Event Simulation, Academic Press, Orlando, FL, 1984.
[15] M. Handley, "Reference simulations for multicast congestion control," IETF RMRG Workshop, London, U.K., 1998; http://www.east.isi.edu/rm/london/mjh.ps.gz
[16] S. Paul, K. K. Sabnani, J. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," IEEE J. Select. Areas. Comm., vol.15, no.3, pp.407-421, April 1997.
[17] D. Wetherall, J. Guttag, and D. L. Tennenhouse, "ANTS: a toolkit for building and dynamically deploying network protocols," Proc. IEEE OPENARCH, San Francisco, CA, pp.117-129, April 1998.



Jong-Kwon Lee received the B.E and M.E degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1994 and 1996, respectively. He is presently a Ph.D candidate at the Department of Electrical Engineering and Computer Science, KAIST. His research interests include network modeling/simulation, congestion control for multicast communication, and active networks. He is a student member of IEEE.



Tag Gon Kim received his Ph.D in computer engineering with specialization in methodology for systems modeling/simulation from University of Arizona, Tucson, AZ, 1988. He was a Full-time Instructor at Communication Engineering Department of Bukyung National University, Busan, Korea between 1980 and 1983, and an Assistant Professor at Electrical and Computer Engineering Department of University of Kansas, Lawrence, Kansas, U.S.A. from 1989 to 1991. He joined at Electrical Engineering Department of KAIST, Daejeon, Korea in Fall, 1991 as an Assistant Professor and has been a Full Professor since Fall, 1998. His research interests include methodological aspects of systems modeling simulation, analysis of computer/communication networks, and development of simulation environments. He has published more than 100 papers on systems modeling, simulation and analysis in international journals/conference proceedings. He is a co-author (with B.P. Zeigler and H. Praehofer) of the book *Theory of Modeling and Simulation* (2nd ed.), Academic Press, 2000. He is the Editor-in-Chief of *Transactions of Society for Computer Simulation* published by Society for Computer Simulation International (SCS). He is a senior member of IEEE and SCS and a member of ACM and Eta Kappa Nu. He was listed in Who's Who in the World (Marquis 16th Edition) in 1999.