PAPER *Special Section on Internet Technology V*

Performance Study and Deployment Strategies on the Sender-Initiated Multicast

Vasaka VISOOTTIVISETH[†*a)], ***Member***, **Hiroyuki KIDO**[†b)], ***Nonmember***, **Katsuyoshi IIDA**[†c)],
Youki KADOBAYASHI[†d)], ***and*** **Suguru YAMAGUCHI**[†e)], ***Members***

SUMMARY Although IP Multicast offers efficient data delivery for large group communications, the most critical issue delaying widespread deployment of IP Multicast is the scalability of multicast forwarding state as the number of multicast groups increases. Sender-Initiated Multicast (SIM) was proposed as an alternative multicast forwarding scheme for small group communications with incremental deployment capability. The key feature of SIM is in its Preset mode with the automatic SIM tunneling function, which maintaining forwarding information states only on the branching routers. To demonstrate how SIM increases scalability with respect to the number of groups, in this paper we evaluate the proposed protocol both through simulations and real experiments. As from the network operator's point of view, the bandwidth consumption, memory requirements on state-and-signaling per session in routers, and the processing overhead are considered as evaluation parameters. Finally, we investigated the strategies for incremental deployment.

key words: *routing, multicast, small group communication, forwarding states reduction*

1. Introduction

IP Multicast [1] was proposed to reduce network bandwidth consumption to less than that required for multiple point-to-point links and to provide efficient data delivery to a large number of receiver. Although, as pointed out in [2], IP Multicast has a high initial cost, ideally the cost per user should decline as the number of new receivers increases. The *sweet spot* of IP Multicast deployment suggested by Cain and described in [2] is the point at which the additional cost of providing the service is outweighed by the benifit of the increased performance.

Examination of the *sweet spot* reveals that IP Multicast provides efficient delivery over multiple unicast and is suitable for *large* group communications. Why then has IP Multicast not been deployed widely, as it should be? Many researchers have attempted to answer this question. As mentioned in [3] by J. Cui et al., the most critical issue delaying the deployment of IP Multicast is the scalability of multicast forwarding state as the number of multicast groups increases.

With respect to scalability, several approaches to aggregating multicast states on routers have been proposed [4]–[6]. Thaler and Handley proposed an interface-centric data structure model that allows aggregation of ranges of multicast addresses in the forwarding table [4]. Rodoslavov et al. proposed algorithms by which to aggregate forwarding state and investigated the bandwidth-memory tradeoff through simulations in [5]. However, it was argued in [6] that both of these studies attempt to aggregate routing state after the distribution trees have been established and that such an implementation is complex. Cui et al. proposed a protocol called *Aggregate Source Specific Multicast* (*ASSM*) [6] to improve the state scalability of Source Specific Multicast. The tradeoff of their approach would be the extra bandwidth required in delivering multicast data to non-group-member nodes.

In addition to the scalability issue, we also examine the usage trend of multicast as another deployment issue. This trend was determined from the multicast statistics data disclosed on web pages, such as data from the CAIDA project [7], and reveals that, at present, most multicast sessions are relatively small, consisting of less than 50 receivers.

There is no *one-size-fits-all* solution to the problems discussed above. For small group communications, the network-related costs per user for IP Multicast is relatively high. "Explicit Multicast (Xcast)" [8], [9] was proposed in order to increase scalability in small group communications. Basically, Xcast routers do not maintain forwarding information and do not exchange control messages with peer routers. Xcast routers forward packets according to the list of receiver addresses attached to the Xcast packet header. Thus, the Xcast sender has to attach the receiver list to every packet, and each on-path Xcast router has to perform routing table lookup for each receiver in the list. That is, Xcast introduces additional processing overhead compared to IP Multicast.

As an alterative to IP Multicast, we have proposed "Sender-Initiated Multicast (SIM)" [10], [11], a single-source protocol supporting small group communications. Small groups are groups consisting of not more than 100 people, which are typically used in video conferencing, collaborative applications, and distance learning. SIM employs the fundamental concept of the packet forwarding mechanism from Xcast, that is, attaching a receiver list to the packet header and making the routers forward and copy the packets for each receiver by looking up a unicast routing ta-

---

Manuscript received June 28, 2004.

[†] The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

[*] Presently, with the Computer Science Department, Mahidol University, Thailand.

a) E-mail: ccvvs@mahidol.ac.th
b) E-mail: hiro-kid@is.naist.jp
c) E-mail: katsu@is.naist.jp
d) E-mail: youki-k@is.naist.jp
e) E-mail: suguru@is.naist.jp

DOI: 10.1093/ietcom/e88–b.4.1383

ble. However, the SIM design goals are increased simplicity compared to the traditional IP Multicast and reduction of header-processing overhead compared to Xcast.

The key feature of SIM is in its Preset mode, which reduces the cost of routing table lookup and provides cost-efficient packet forwarding through SIM Forwarding Information Base (FIB) maintained on routers. Moreover, automatic SIM tunneling yields scalability by maintaining FIB entries only on the branching routers. SIM tunneling is especially useful for applications such as teleconferencing, in which the members of multicast groups are usually sparsely distributed in the network, and the probability that the packet has to be duplicated or branched on the same router is low. The SIM Tunnel mechanism accelerates the forwarding process and potentially reduces the number of routers that must maintain SIM FIB.

To demonstrate how SIM increases scalability with respect to the number of groups, in this paper we compare the performance of SIM in the Preset mode with the performances of Xcast for IPv4, PIM-SM and unicast through simulations and experiments. As from the network operator's point of view, the bandwidth consumption, memory requirements on state-and-signaling per session in routers, and the processing overhead are considered as the evaluation metrics. In addition, since SIM requires at least one SIM-enabled router in the Internet, we investigate the strategies for incremental deployment.

The rest of the paper is organized as follows. Section 2 gives an overview of SIM, and Sect. 3 explains SIM in detail with respect to tree maintenance. Section 4 presents the performance evaluation methods and the results of the simulations and experiments. Section 5 discusses gradual deployment issues. Section 6 presents related work, and the present study is summarized in Sect. 7.

2. Overview of SIM

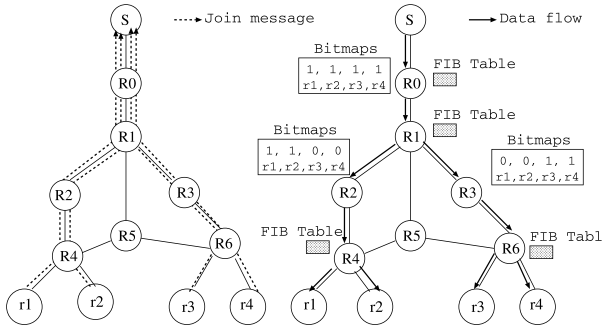
Originally, SIM has been proposed in [10] and [11]. In this section and the next section, we introduce SIM before presenting our performance evaluation. If you are familiar with SIM, you can skip Sects. 2 and 3 and go to Sect. 4.

2.1 SIM Membership Management

Similar to Xcast, we assume that the SIM sender knows all of the receiver addresses in advance before beginning transmission. SIM has no join or leave messages traversing through routers. Each receiver sends subscription/unsubscription data directly to the sender through email or a web page (Fig. 1(a)). The sender can find the IP addresses of receivers by various methods, such as email or a Web browser.

2.2 SIM Forwarding Techniques

There are two SIM forwarding modes: *List mode* and *Preset mode*.



(a) Each receiver send join message to the source

(b) Data flow creates FIB entries on branching routers

Fig. 1 Multicast tree construction.

- **List mode**
  List mode is appropriate for short data transmission, retransmission and cases in which the receiver list is supposed to be modified frequently. The sender will attach the receiver list to each packet. SIM routers do not maintain any forwarding information, i.e. SIM FIB entries. In addition, like Xcast routers, SIM routers must look up the unicast routing table for each receiver each time the packets arrive.
- **Preset mode**
  This mode is appropriate for long data transmission and cases in which the receiver list is seldom modified. The sender will periodically attach the receiver list to only some of the packets. SIM routers maintain a SIM FIB for each multicast group. The most attractive characteristic of the Preset mode is the SIM Tunnel. SIM Tunnels are automatically created among the branching-point SIM routers. Only routers at branching-points of the distribution tree must maintain SIM FIB entries and process SIM packet forwarding.

In both forwarding modes, the sender attaches a bitmap field to each packet. The bitmap concept was originally proposed in Connectionless Multicast (CLM) [12] and is taken into consideration in the basic specifications of Xcast [8]. Each bit in the bitmap field must correspond with the sequence of the receiver list, e.g. the first bit corresponds to the first destination in the list (Fig. 1(b)). The bitmap is used in order to abbreviate the receiver list, when the packet has to be forwarded to the same set of receivers. Moreover, when the router has to duplicate the packet, complex packet-header reconstruction can be avoided by simply turning on or off the bitmaps according to the routes to each receiver.

In order to clarify the significant characteristics of SIM, we will herein discuss only SIM in the Preset mode.

2.3 Conversion-to-Unicast

SIM conversion-to-unicast is performed in two cases: the case in which there is no SIM-capable router on the next hop and the case in which there is only one receiver for the next hop router. The SIM router detaches the SIM header and its options and then rewrites the source and destination address fields of the packet. If necessary, the SIM router also rewrites some of the fields of the upper-layer protocol header, such as checksums and port numbers. These substitution data are created by the sender and are stored in a SIM option header in the sequence of the receivers in the address list. As a result, the packet appears to originate from an ordinary unicast application. Thus, all receivers can receive data without modification of the existing operating system or applications. Delivery by TCP can also be supported [13].

3. SIM Tree Maintenance

Multicast tree maintenance is a crucial issue affecting the protocol scalability. In this section, we present two mechanisms using in maintenance the SIM forwarding path. The first one is the management of SIM Forwarding Information Base on routers. The second one is SIM Tunnel by using SIM Redirect message.

3.1 Management of SIM FIB

Despite performing routing table lookup for each receiver in all packets, each SIM router maintains a hash table of the Forwarding Information Base (FIB) for each multicast session. The hash value of each session is calculated from the sender address and multicast group address. The SIM FIB entry is registered as three tuples: the generation counter, the incoming network interface (or the previous branching router when SIM Tunnel is used), and the outgoing interface (or tunnel egress address when a SIM Tunnel is used).

The generation counter field in the SIM header is used to enable routers on a multicast tree to detect changes in the address list. Whenever the receiver list of a flow changes, for example due to a receiver joining or leaving the group, the generation counter will be incremented. In addition, the initial value of the generation counter is created randomly in order to avoid a Denial-of-Service (DoS) attack.

SIM FIB entries are soft state. In addition, the sender increase the generation counter and attaches a receiver list to the packets for each *SIM_FIB_INTERVAL* interval time. Even if no appropriate packet to attach a receiver list exists, a SIM packet without data will be sent. SIM routers will update the SIM FIB entry each time a new generation counter is received. In contrast, if the SIM routers do not receive a receiver list for a data flow for the *SIM_FIB_TIMEOUT* time, the SIM routers will remove the corresponding SIM FIB entry. However, the SIM FIB entry of an old generation counter should be kept for *SIM_FIB_AGING* time, since some packets with the same generation counter may be delayed by network congestion or network failure.

When a sender sends the last packet of a flow, the sender can request SIM routers to remove the corresponding SIM FIB entries before the *SIM_FIB_TIMEOUT* timer expires by turning on the *Delete flag* in the packet header. The routers then set a *SIM_FIB_AGING* timer and remove the entry after the timer expires.

3.2 SIM Tunnel and Redirect Message

The automatic SIM Tunnel is a significant feature of SIM. The advantage here is that only the branching routers have to maintain the SIM FIB entries and perform routing table lookup for each receiver on the list.

In Preset mode, there are two cases in which a branching router sends a *SIM Redirect Message* to the upper router:

- *To automatically create a SIM Tunnel between two routers that act as branching points of a multicast distribution tree*
  Whenever packets pass through a non-branching SIM router, the router will turn on the *Jump flag* on the packet header (Fig. 2). The next SIM branching router will detect this flag when the packet arrives and send a SIM Redirect Message to the previous SIM branching router to create a SIM Tunnel (Fig. 3).

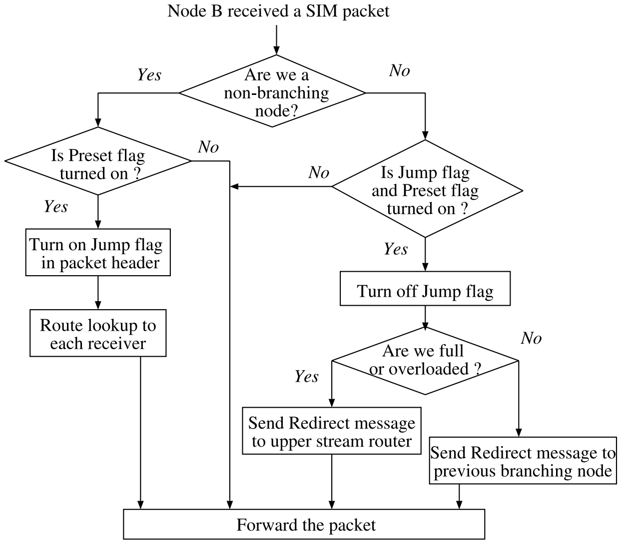


Fig. 2 Jump flag processing.

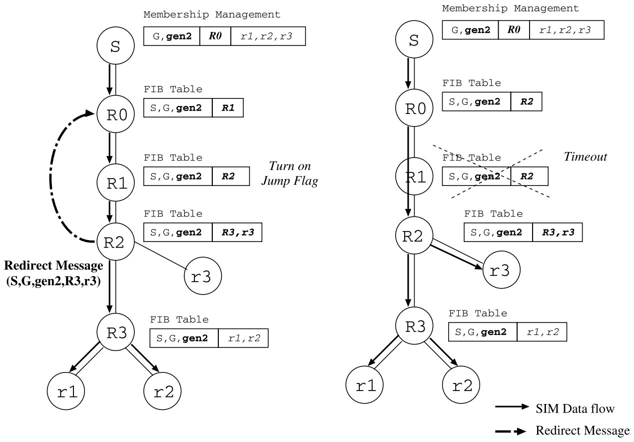


Fig. 3 SIM Redirect message and SIM Tunnel for capable new branching node.

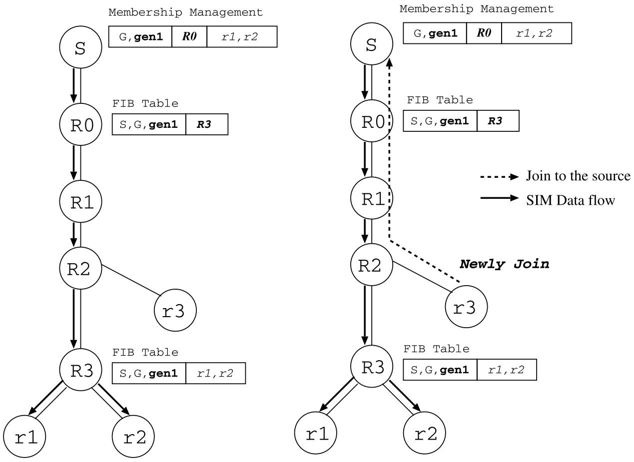


Fig. 4 New branching node by newly joined receiver.

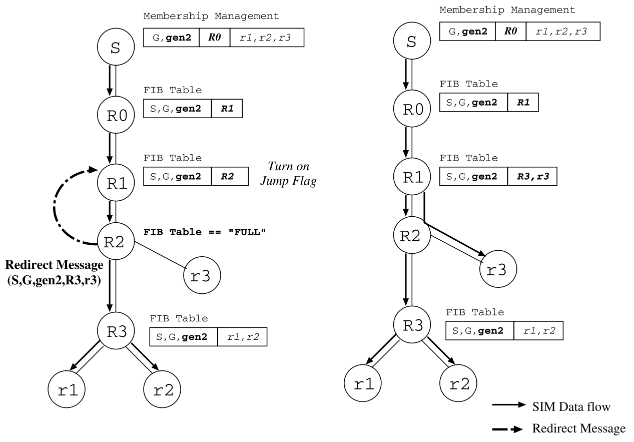


Fig. 5 SIM Redirect message for overloaded node.

- *To request an upper router to pre-duplicate packets, when the router sending the Redirect Message is overloaded*
  In Fig. 4, R2 is a new branching router created as the receiver r3 begins to subscribe to the group. Normally, if R2 is not overloaded, R2 will send a SIM Redirect Message to create a SIM Tunnel. However, when the SIM FIB hash table of the router is full, i.e. the router cannot maintain more FIB entries but can still forward the packets, the router sends a SIM Redirect Message to tell the upper router R1 of the data flow to duplicate packets in advance (Fig. 5).

Normally, when a SIM packet is transferred through the SIM Tunnel, the packet must be encapsulated using another IP header with the source address pointing to the originating router and destination address pointing to the egress router. Note that the source address of this encapsulated header will be changed only when the packet passes through a SIM branching router. Therefore, the next branching SIM router can determine the previous branching SIM router from the source address field of the packet. When the previous branching router receives the SIM Redirect message, it updates the appropriate SIM FIB entry and specifies the next SIM branching router in the destination address field of the packet to create a SIM Tunnel. Thus, non-branching routers will recognize SIM packets as ordinary unicast packets, and simply route them to the next branching router.

4. Performance Evaluation

From the network operator's point of view, the bandwidth consumption, the router's processing overhead, and the memory requirements on state-and-signaling per session in routers are considered as the important factors affecting the scalability of network protocols.

4.1 Evaluation Methods: Simulation and Experiment

In order to investigate how SIM increases the scalability with respect to the number of groups, we implemented SIM prototype for IPv4 as a kernel option with kernel modifications for FreeBSD 4.6-RELEASE. However, only experiments on the implementation of the protocol using a network of limited scale and the number of hosts cannot accurately show how SIM scales the number of groups on the actual Internet. Therefore, we have also developed a simulator [14] by using C++ language. The simulator consists of SIM routers acting in either the Preset mode or the List mode. Each SIM router maintains a static routing table with the knowledge of all destinations.

We compare through simulations the SIM performance with the respective performances of PIM-SM and Xcast for IPv4 (Xcast4), excluding the packet forwarding time in Sect. 4.3.1, which is measured through experiments. We chose PIM-SM and Xcast4 because PIM-SM has a relatively long history and is currently deployed, also forwarding according to the list of receiver addresses as in Xcast is the fundamental concept of SIM. In addition, the existing implementation of Xcast4 is based on the Linux kernel. Therefore, in this paper we use SIM in the List mode as a substitute for Xcast4.

Before we discuss the performance evaluation results obtained by the both implementation and simulations in the next section, we first validate that both the simulator and the implementation produce the same results. Our validation is performed using the network topology, referred to as *Hop-count* topology, depicted in Fig. 6. This topology consists of a sender, 31 routers, and 30 receivers. All nodes are 1-GHz Pentium CPUs with 512 MB of memory and are well connected by 100 Base-T Ethernet links. Note that the maximum number of receiver nodes in this experiment is limited by the number of available computers.

The position of the branching points is important for the performance evaluation of SIM, as SIM Tunnels are created among the branching points. The longer the tunnel, i.e. the distance between the branching nodes, the better the performance of SIM becomes. Therefore, the topology is based on the average hop counts, as described in [15], which reveals that the IP-path length, i.e. the number of hop counts

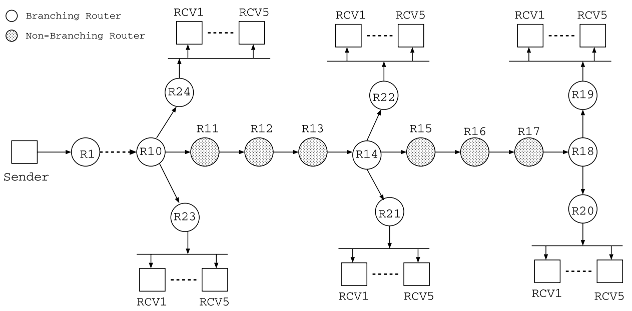


Fig. 6 Topology used in validation of the simulator and the implementation.

Table 1 Total number of routing table lookups observed from the simulation and experiment.

Number of receivers	Number of Lookups	
	SIM	Xcast
6	288	1080
12	288	2160
18	288	3240
24	288	4320
30	288	5400

required for most senders to reach each receiver is approximately 15 ± 4 hops. Thus, in multicast transfer, the 11th, 15th, and 19th routers are usually the branching points of the distribution tree.

In both the experiments and simulations, we increase the number of receivers from 6 to 30. In order to fix the position of the branching points even when the number of receivers increases, in each experiment and simulation, the sender sends 12 packets to 6, 12, 18, 24, and 30 receivers, respectively. In other words, in each experiment, each last hop router has 1, 2, 3, 4, and 5 receivers. The results for total routing table lookups on all routers are listed in Table 1.

The results in Table 1 indicate that the number of routing table lookups by SIM is constant regardless of the number of receivers, because SIM Tunnels are created between branching nodes, and all SIM branching nodes maintain SIM FIB. A branching node requires only one lookup for each packet, while a non-branching node considers the packet as a unicast packet and performs only one lookup and one forwarding. On the other hand, Xcast routers do not maintain the forwarding state. For Xcast routers, the number of routing table lookups increases as the number of receivers increases. Since, non-branching nodes in Xcast models also have to perform routing table lookups, the total number of routing table lookups in Xcast model increases remarkably when the number of receivers increases. Finally, the results form the table show that simulations and experiments from the implementation produce the same value. Therefore, we can conclude that both the simulator and the implementation forward packets in the same manner.

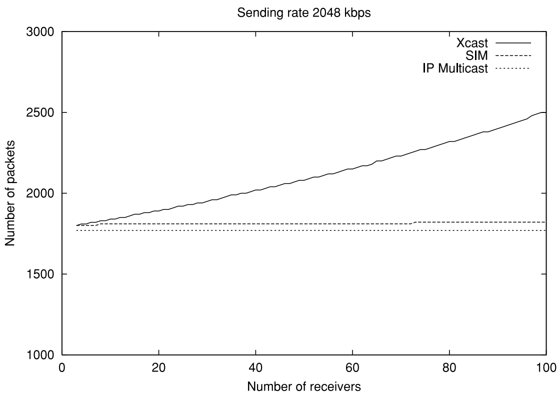


Fig. 7 Number of receivers and sending packets for sending rate 2048 Kb/s.

4.2 Bandwidth Consumption

The bandwidth consumption is both the additional bytes increased by the packet header and also the number of duplicated packets in each network link. However, since we assume that all compared protocols use the shortest path forwarding and produce non-duplicated packets, the latter can be ignored. Therefore, bandwidth consumption can be considered as the overhead of a packet header in Xcast4 and SIM, as both require an additional newly defined header attached to the packets. In contrast, the conventional IP Multicast only requires the sender to specify a multicast group address as a destination. Hence, IP Multicast produces no overhead in bandwidth consumption compared to the ordinary unicast.

We evaluate bandwidth consumption based on the number of packets the sender has to transmit in order to achieve a certain data rate. We assume that the sender transmits 2048 Kb/s data flows to 3 to 100 receivers using these three protocols. In order to make a distinction between SIM in the Preset mode and Xcast4, we set the total sending time to 10 seconds, which is equal to the default *SIM_FIB_INTERVAL* time, and the path MTU to all receivers as 1500 bytes. The simulation results are shown in Fig. 7.

We found that at a sending rate of 2048 Kb/s for 50 receivers, Xcast4 and SIM have to send 17.5% and 2.3% more packets, respectively, compared to IP Multicast. For 100 receivers at the same rate, the overheads are 41.2% and 2.8%. Therefore, attaching a receiver list to every packet, as in the Xcast4 model, introduces more bandwidth consumption than SIM in the Preset mode. This overhead becomes larger when the sending rate and number of receivers in a group increase. Note that, since the SIM sender always attaches a receiver list for each *SIM_FIB_INTERVAL* time, the number of packets sent in the y-axis direction will gradually increase according to the duration of the simulation.

4.3 Packet Processing Overhead

Packet processing overhead on routers can be observed from the forwarding time in the real data transfer. Moreover, this overhead is caused by packet duplication and route computation (routing table lookup). However, packet duplication occurs only on multicast branching routers, and is the same for all three protocols: SIM, PIM-SM, and Xcast4. Therefore, the route computation overhead is the obvious factor influencing the packet processing overhead.

4.3.1 Packet Forwarding Time

The packet forwarding overhead is a consequence of the processing in the SIM and Xcast forwarding routines. We measure the average packet forwarding time in the experiments for the network topology described in Sect. 4.1. In order to measure with high precision, we measure the number of CPU cycles changed after each forwarding process by using the read timestamp counter (RDTSC) instruction. The forwarding time is calculated by dividing the number of CPU cycles by the CPU clocks in Hertz. Table 2 shows the average packet forwarding time on branching routers for SIM and Xcast4 in μs.

As shown in Table 2 the SIM average forwarding time remains approximately the same and is not affected by the number of receivers. In contrast, for Xcast, the greater the number of receivers, the more the average forwarding time increases. From this experimental result, we can confirm that the most important factor that causes SIM end-to-end delay, is not the overhead of routing table lookup, but rather packet duplications on the last hop router.

Table 2 Average forwarding time of SIM and Xcast on branching routers (μs).

Number of receiver	SIM	Xcast
6	6,312	1,425,998
12	6,382	1,486,012
18	6,412	1,500,808
24	6,751	1,510,390
30	6,814	1,548,267

4.3.2 Route Computation

In order to compare route computation overhead, we measure the number of routing table lookups for each protocol. However, IP Multicast uses a multicast address as group identification, and thus the number of routing table lookups should be the same as for SIM in the Preset mode. Therefore, we measure only the number of routing table lookups for Xcast4 and SIM.

In order to determine how to achieve scalability of SIM with respect to the number of groups, we perform simulations using the actual topology of the Japan Gigabit Network (JGN) (Fig. 8) [16]. The JGN is a nationwide network in Japan designed for the research and development of very-high-speed networking and high-performance application technologies. This network consists of 10 backbone routers connected by gigabit network links and approximately 60 routers connected to thick links of 50 Mbit/s to 2.4 Gbit/s. We assume that each router, excluding backbone routers, connects with four end nodes. Therefore, a total of 240 end nodes act as multicast senders and receivers in these simulations. Routing paths are assumed being symmetries. Here, we assume small multicast sessions, such as video conferencing or network games among only a few people. These types of sessions can be held using multiple one-to-many multicast sessions.

In our simulations, we fix the number of receivers to four and varied the number of multicast groups from 10 to 700. Thus, in total, we have 40 to 2800 multicast sessions. The sending rate is 500 Kb/s. The results of the simulation are shown in Fig. 9. From these results, we found that

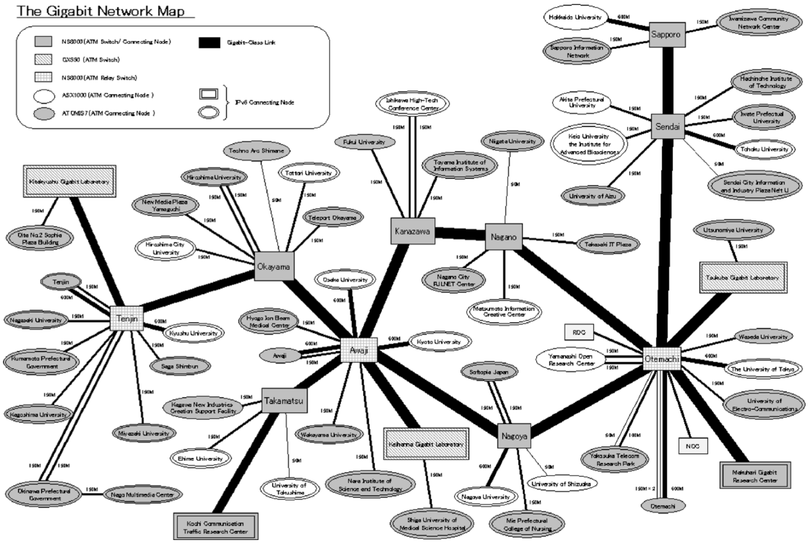


Fig. 8 Japan gigabit network topology.

Xcast4 requires approximately 1.5 times as many routing table lookups as SIM. This is because SIM has a significant scheme, SIM FIB and SIM Tunnel, which reduces the number of routing table lookups and the number of FIB entries maintained on each router. The simulation results provide evidence that SIM reduces packet processing overhead and provides scalability to a large number of small groups.

4.4 State-and-Signaling per Session

The state-and-signaling cost of each protocol depends on the implementation. Concerning to the state cost, Billhartz et al. argued in Sect. 4.5 of [17] that the intricacy of the protocol, operating system overhead, and routing table size become especially important when the number of senders and groups is large, because router speed and memory requirements are influenced. For the quantitative evaluation, we choose the routing table size at each router as the performance metrics.

Since Xcast4 does not require each router to maintain a table of multicast routing information, we compare only the routing table size of PIM-SM and SIM. In addition, as a backbone router has more chances to be a branching point, we focus on the routing table size on each backbone router. We perform simulations with the same JGN network topology depicted in Fig. 8, and examine the relationship between the number of groups and the routing table size. We fix the number of receivers to four, and vary the number of multicast groups from 10 to 10,000. Therefore, in the simulations, the number of multicast sessions varies from 40 to 40,000. Each sender sends only one packet to the group. On 10 backbone routers, for each protocols, we observe parameters described in Table 3.

Note that a PIM-SM routing entry S_{rt} consists of information of source, group, and upstream router. The number of packet duplication N_d represents the number of downstream entries required on each router. As PIM-SM in our simulation constructs source-based trees for all (S,G) session, the average size of routing table on per router S_{pimsm} is given by the following expression.

$$S_{pimsm} = S_{rt} \cdot N_{all} + S_{fe} \cdot N_{br} + B_{pim} \cdot N_d.$$

On the contrary, a SIM router maintains SIM FIB entries only if the router is a branching point. Therefore, the average size of routing table per router S_{sim} is given by the following expression.

$$S_{sim} = S_{fib} \cdot N_{br} + B_{sim} \cdot N_d.$$

To illustrate how the average size of routing table increases with the number of groups, we input values N_{all}, N_{br} and N_d obtained from the simulation. The sizes of a routing entry S_{rt}, multicast forwarding cache entry S_{fe}, downstream entry B_{pim}, and the sizes of a SIM FIB entry S_{fib}, SIM destination entry B_{sim} are obtained from the header files of each implementation [18], [19]. We found that S_{rt}, B_{pim}, S_{fe}, S_{fib}, and B_{sim} are respectively 148, 32, 74, 88, and 36 bytes. Figure 10 shows the growth of memory required on a backbone router with the growth in the number of groups.

Moreover, from Fig. 10, we can see relationships between the number of groups and an approximate size of routing table for each protocol as follows:

$$S_{pimsm} \approx 505 \cdot G, \tag{1}$$

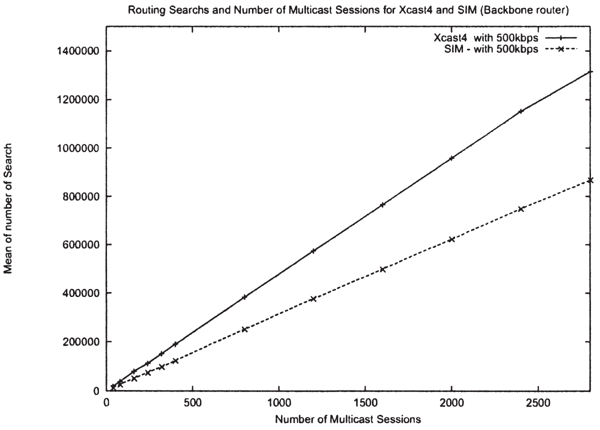


Fig. 9 Routing table search and the number of sessions.

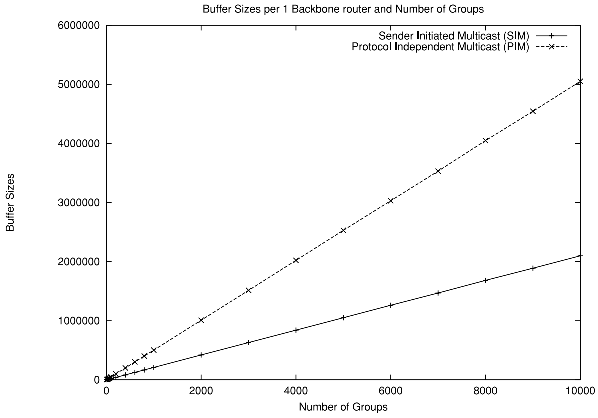


Fig. 10 Routing table size (Bytes) on a backbone router with the growth in the number of groups.

Table 3 Parameters observed in the simulation for routing table size.

N_{all}	Total number of all sessions (S,G) passing through a router
N_{br}	Total number of sessions (S,G) performed packet duplication at the branching router
N_d	Total number of packets duplication performed on the router
S_{rt}	Size (Bytes) of a PIM-SM routing entry for a (S,G) session
S_{fe}	Size (Bytes) of a multicast forwarding entry in the PIM-SM kernel
S_{fib}	Size of a SIM FIB entry with a generation number for a (S,G) session
B_{pim}	Size (Bytes) of a PIM-SM downstream entry for a (S,G) session
B_{sim}	Size of a SIM downstream entry for a (S,G) session

$$S_{sim} \approx 210 \cdot G. \tag{2}$$

To illustrate how many groups each protocol can support, we give the maximum memory size of a router to Eqs. (1) and (2). For example, assuming the memory size on a backbone router is 128 MBytes, an approximate maximum number of multicast groups a PIM-SM and a SIM router can support are 253,465 and 609,524, respectively. We can conclude that, on average, each SIM router requires approximately 41% of the routing table size required by a PIM-SM router.

5. SIM Gradual Deployment

Another issue delaying the deployment of multicast besides the scalability issue is the complexity of deployment. With respect to the gradual deployment of SIM, we need to determine where to place SIM routers on the Internet, how many SIM routers are required, and how the packet processing overhead changes when the number of network branches increases.

To simplify our problems, we assume that the Internet is a mixed network of the three types of network topologies: *bus*, *tree*, and *star* topologies, while the bus and tree topologies normally represent router connections, and the star topology represents an inter-domain router with multiple network branches.

First, in order to address the first and second questions, we perform two experiments for each bus and tree topology. Figures 11 and 12 show the bus and tree topologies, respectively, used in our experiments. Each topology consists of a sender, 31 routers, and 32 receivers. All nodes are well connected to each other by 100 Mbit/s link. In each experiment, we observe the the total routing table lookups when the sender transmits 12 packets to 32 receivers. Note that the worst case for SIM is when every router becomes a branching point of the distribution tree because SIM cannot create a SIM Tunnel. Moreover, the number of receivers next to each router is not the factor that must actually be considered.

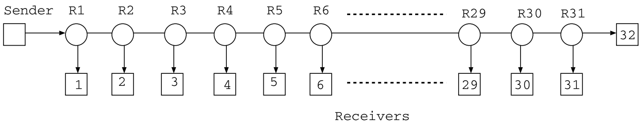


Fig. 11 Bus topology.

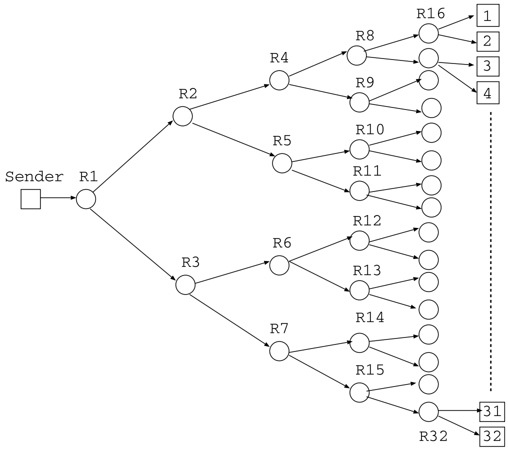


Fig. 12 Tree topology.

5.1 Placement of SIM Routers on the Internet

For the initial deployment, since SIM requires at least one SIM-enabled router in the Internet, where should we place the first SIM router? We assume that there is only one SIM router in each topology. The location at which to place the SIM router changes gradually from the first router to the last router. As shown by the results in Sect. 4.3, since the end-to-end delay is proportional to the number of routing table lookups, measuring only the number of routing table lookups is sufficient in order to evaluate the proposed protocol. Figures 13 and 14 depict the total routing table lookups in the bus and tree topologis, respectively.

Figure 13 shows that when a SIM router is the second router of the bus topology, the total number of routing table lookups increases slightly. This is because the duplicated packets have been forwarded back to the first router.

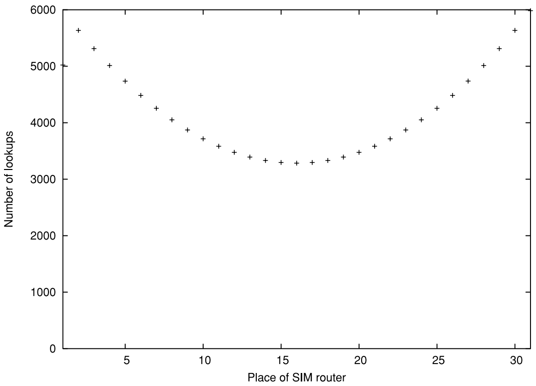


Fig. 13 Total routing table lookups in Bus topology, while only one SIM router exists.

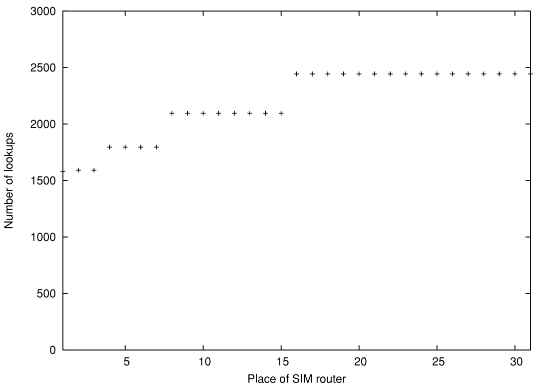


Fig. 14 Total routing table lookups in Tree topology, while only one SIM router exists.

However, when the SIM router is located near the center of the network, the total number of routing table lookups decreases. This result is also due to duplicated packets passing through each network link. At the point of the 15th and 16th routers, duplicated packets are equally forwarded to the previous and next routers. Therefore, the total number of routing table lookups is the least in this experiment.

For the tree topology, Fig. 14 shows the total number of routing table lookups increases gradually. The further the SIM router is located from the sender, the greater the requirement for routing table lookup. The number of duplicated packets is the same for routers having equal hop length from the sender. We therefore conclude that, for bus topology, the best place to deploy SIM is at the center of a network; whereas for tree topology, the best place to deploy SIM is the router nearest the sender node.

5.2 Number of SIM Routers Required

In order to determine the number of SIM routers required, we performed the second experiment using the bus and tree topologies described above. In contrast to the first experiment, we deploy multiple SIM routers gradually from the nearest router to the furthest router, when viewed from the sender. Figures 15 and 16 depict the total number of routing table lookups for each topology.

Both figures show that when the number of SIM routers increases, the total number of routing table lookups gradually decreases. This means that the greater the number of SIM routers deployed, the greater the reduction in packet processing overhead. Moreover, neither of the slopes in the graphs decreases linearly. The total number of routing table lookups decreases slightly when all routers deployed SIM. In other words, the greatest benefit is obtained if half of all on-path routers deploy the proposed protocol.

5.3 Change in Packet Processing Overhead, When the Number of Network Branches Increases

In order to investigate the change in packet processing overhead when the number of network branches increases, we perform an experiment using a star topology, since the number of network branches can be represented by the number of receivers. Figure 17 depicts the Star topology consisting of a sender, a SIM router, and six nodes acting as receivers. Since each receiver node is alias to 10 IP addresses, we have a total of 60 receivers. The number of network branches represented by the number of receivers gradually increases from 6 to 58 during this experiment, where 58 is the maximum number of receivers supported by the present SIM implementation. The measurement parameter is the average forwarding time on the router, which can be considered as the router overhead. In this experiment, the number of routing table lookups is obviously equal to the number of receivers and therefore is not measured.

Table 4 shows the experimental results measured on a

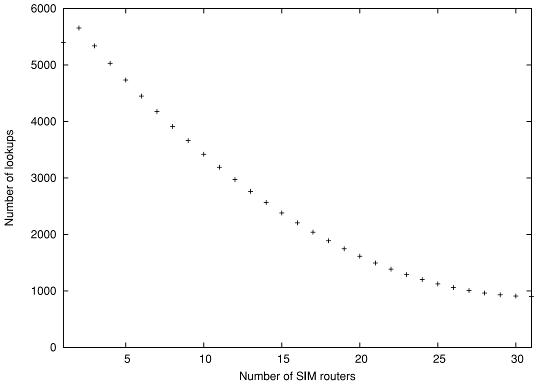


Fig. 15 Total routing table lookups in Bus topology, while number of SIM routers increases.

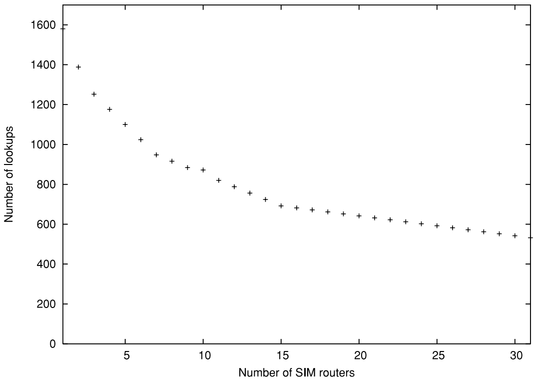


Fig. 16 Total routing table lookups in Tree topology, while number of SIM routers increases.

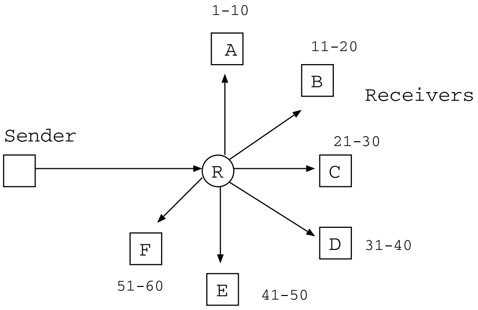


Fig. 17 Star topology.

Table 4 Average forwarding time on a branching router in Star topology (μs).

Number of receivers	Forwarding Time (μs)
6	52,956
12	104,484
18	140,096
24	154,742
30	175,503
36	203,044
42	243,524
48	261,399
54	310,562
58	339,251

computer (450 MHz CPU and 128 MB of memory) acting as our SIM router. The results from this table show that the average forwarding time increases as the number of network branches increases. However, this increment is sublinear with respect to the number of receivers, as indicated by the fact that when the number of receivers is 54, the average time is only approximately six times greater than that for six receivers.

6. Related Work

EXPRESS [20] is a single source multicast protocol extended from IP Multicast to support explicit subscription, and to provide membership information to the source. A simple solution for address allocation and management was resolved by identifying the tuple of (S,G) as a multicast channel, where S is the unicast address of the source and G is the address in the range of 232/8. Source Specific Multicast (SSM) [21], [22] or PIM-SSM is the successor to EXPRESS, and is implemented by extending the conventional IP Multicast (PIM-SM). SSM alleviates a number of problems other than the scalability issue in terms of the number of groups, such as access control and address allocation. Simple Multicast [23] is a core-based multicast routing protocol that also extended the existing group identification. The group identification is (C,G), where C is the core router address and G is the multicast group address. SIM adopts this concept of group identification in order to eliminate the multicast address allocation problem.

The aggregated multicast approach [3] was designed to eliminate forwarding states on routers inside a single transit domain which has no branching points on the multicast tree. However, both the ingress and egress routers of the domain still have to maintain forwarding states. The aggregated multicast approach is similar to the proposed SIM protocol in that both approaches sacrifice tunneling cost. The difference is that, SIM can also eliminate forwarding states on the ingress and egress routers.

Dynamic Tunnel Multicast [24] was proposed to achieve state reduction at non-branching routers in sparse communication. Similar to SIM, Dynamic Tunnel Multicast dynamically sets up tunnels between branching routers. As argued in [25], in order to support dynamic membership, Dynamic Tuennl Multicast requires a more sophisticated control protocol to set up tunnels and to deal with route changes.

Like the proposed protocol, REcursive UNIcast TreE (REUNITE) [25] is based on unicast routing infrastructure. However, REUNITE does not consider asymmetric routing, and may therefore fail to construct the shortest paths, or may produce unneeded packet duplication on certain links. The asymmetric routing problem in REUNITE has been resolved by the HBH (Hop-By-Hop multicast routing protocol) [26]. A drawback of HBH is that the situation in when the router is overloaded is not considered.

7. Conclusions

Sender Initiated Multicast (SIM) was designed to support small group communications with a cost-efficient packet forwarding mechanism and a significant deployment strategy. The design goal for SIM is not to replace the conventional multicast protocol, but to provide SIM as an alternative subset of the multicast protocol suite. The disadvantages of the proposed protocol are the constraint in the number of receivers and the consumption of memory resources on branching routers.

In this paper, we investigated how SIM increase scalability with respect to the number of groups through simulations and experiments. The studied results revealed that SIM can reduce the bandwidth consumption and packet processing overhead compared to the Xcast protocol and has less state-and-signaling per session compared to the IP Multicast. For gradual deployment, we moreover observed that, for bus topology, the best place to deploy SIM is at the center of a network; whereas for tree topology, the best place to deploy SIM is the router nearest the sender node. In addition, the greatest benefit of SIM is obtained when half of all on-path routers deploy the proposed protocol. Finally, the processing overhead on routers increases sublinearly with respect to the number of network branches. Direction for future work includes adopting the aggregated multicast scheme for further further router-state reduction among multiple multicast groups.

References

[1] S.E. Deering, "Host extensions for IP multicasting," RFC 1112, Internet Engineering Task Force, Aug. 1989.

[2] C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," IEEE Netw., vol.14, no.1, pp.78–88, Jan./Feb. 2000.

[3] J. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla, "Aggregated multicast—A comparative study," Journal of Networks, Software and Applications, the special issue of Cluster Computing, Baltzer Science Publisher, vol.8, no.1, pp.15–26, 2003.

[4] D. Thaler and M. Handley, "On the aggregatability of multicast forwarding state," Proc. IEEE INFOCOM 2000, vol.3, pp.1654–1663, 2000.

[5] P. Radoslavov, D. Estrin, and R. Govindan, "Exploiting the bandwidth-memory tradeoff in multicast state aggregation," Technical Report 99-697, Dept. of Computer Science, USC, Feb. 1999.

[6] J. Cui, D. Maggiorini, J. Kim, K. Boussetta, and M. Gerla, "A protocol to improve the state scalability of source specific multicast," Proc. IEEE Globecom2002, vol.2, pp.1899–1904, 2002.

[7] M. Project and CAIDA.org, "Session and membership monitor, monitoring characteristics of multicast sessions and participant hosts," March 2002. http://www.caida.org/tools/measurement/Mantra/session-mon/

[8] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, "Explicit multicast (Xcast) basic specification," Internet-Draft draft-ooms-xcast-basic-spec-05.txt, Internet Engineering Task Force, Aug. 2003. Work in progress.

[9] Y. Imai, H. Kishimoto, M. Shin, and Y. Kim, "XCAST6: eXplicit Multicast on IPv6," Proc. IPv6 and Applications Workshop in conjunction with IEEE/IPSJ SAINT2003, pp.238–243, Jan. 2003.

[10] V. Visoottiviseth, H. Kido, K. Iida, Y. Kadobayashi, and S. Yamaguchi, "Sender-initiated multicast forwarding scheme," Proc. 10th International Conference on Telecommunications, pp.334–339, Feb./March 2003.

[11] V. Visoottiviseth, K. Kadobayashi, and S. Yamaguchi, "SIM: An efficient multicast forwarding scheme for small group communications," IEICE Trans. Commun. (Japanese Edition), vol.J85-B, no.8, pp.1215–1226, Aug. 2002.

[12] D. Ooms and W. Livens, "Connectionless multicast," Internet-draft, Internet Engineering Task Force, April 2000. draft-ooms-cl-multicast-02.txt, Work in progress.

[13] V. Visoottiviseth, T. Mogami, and S. Yamaguchi, "M/TCP: Multicast extension to TCP," Proc. ICACT 2001, Muju, Korea, pp.89–94, Feb. 2001.

[14] H. Kido, Performance evalution of router processing load and group scalability in explicit multicast, Master's Thesis, NAIST-IS-MT0051028, Nara Institute of Sceince and Technology, Feb. 2002.

[15] B. Huffaker, M. Fomenkov, D.J. Plummer, D. Moore, and K. Claffy, "Distance metrics in the Internet," Proc. IEEE International Telecommunications Symposium (ITS), 2002. http://www.caida.org/outreach/papers/2002/Distance/distance.pdf

[16] J.G. Network, "Topology of Japan gigabit network," http://www.jgn.nict.go.jp/english/03-network/3-2.html

[17] T. Billhartz, J.B. Cain, E. Farrey-Goudreau, D. Fieg, and S. Batsell, "Performance and resource cost comparison for the cbt and pim multicast rouging protocols," Proc. IEEE INFOCOM'96, vol.1, pp.85–93, San Francisco, USA, 1996.

[18] U. of Southern California, "PIM in gated (5.0 alpha)," http://www.isi.edu/~eddy/pim/pim.html

[19] U. of Southern California, "PIM-DM (standalone pim-sm with patch)," http://www.isi.edu/~eddy/pim/pim.html

[20] H.W. Holbrook and D.R. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications," Proc. ACM SIGCOMM'99, pp.65–78, 1999.

[21] E.S. Bhattacharyya, "An overview of source-specific multicast (SSM)," RFC 3569, Internet Engineering Task Force, July 2003.

[22] K.C. Almeroth, S. Bhattacharyya, and C. Diot, Challenges of Integrating ASM and SSM IP Multicast Protocol Architectures, Lecture Notes in Computer Science, vol.2170, Springer-Verlag, 2001.

[23] R. Perlman, C. Lee, T. Ballardie, J. Crowcroft, Z. Wang, T. Maufer, C. Diot, J. Thoo, and M. Green, "Simple multicast: A design for simple, low-overhead multicast," Internet-draft, Internet Engineering Task Force, March 1999. draft-perlman-simple-multicast-03.txt, Work in progress.

[24] J. Tian and G. Neufeld, "Forwarding state reduction for sparse mode multicast communication," Proc. IEEE INFOCOM'98, pp.711–719, April 1998.

[25] I. Stoica, T.S.E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," Proc. IEEE INFOCOM 2000, vol.3, pp.1644–1653, 2000.

[26] L. Henrique, M. Costa, S. Fdida, and O.C.M. Duarte, "Hop-by-hop multicast routing protocol," Proc. ACM SIGCOMM 2001, pp.249–259, Aug. 2001.

[27] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol independent multicast-sparse mode (PIM-SM): Protocol specification," RFC 2362, Internet Engineering Task Force, June 1998.

[28] S. Banerjee and B. Bhattacharjee, "A comparative study of application layer multicast protocols," 2002. http://citeseer.ist.psu.edu/banerjee01comparative.html

[29] Y. Chu, S.G. Rao, and H. Zhang, "A case for end system multicast," Proc. ACM SIGMETRICS 2000, pp.1–12, Santa Clara, CA, June 2000.

[30] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," Technical Report, UMIACS TR-2002, 2002. http://citeseer.ist.psu.edu/banerjee02scalable.html

[31] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: A framework for delivering multicast to end users," Proc. IEEE INFOCOM 2002, pp.1366–1375, June 2002.

[32] P. Francis, "Yoid: Your own internet distribution," March 2001. http://www.isi.edi/div7/yoid/

[33] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast infrastructure," Proc. 3rd USENIX Symposium on Internet Technologies and Systems (USITS), pp.49–60, 2001.

[34] J. Widmer and M. Handley, "TCP-friendly multicast congestion control (TFMCC): Protocol specification," Internet draft, draft-ietf-rmt-bb-tfmcc-02.txt, work in progress, July 2003.

Vasaka Visoottiviseth received B.E. and M.E. in Computer Science from Tokyo University of Agriculture and Technology (TUAT), Tokyo, Japan in 1997 and 1999, respectively, and Ph.D. in Information Science from Nara Institute of Science and Technology (NAIST), Nara, Japan in 2003. Since April 2003, she has been a lecturer in the Computer Science Department, Faculty of Science, Mahidol University, Thailand. Her current research interests include multicast, routing, IPv6, traffic measurement and network security. She is a member of the WIDE project in Japan, and the Institute of Electrical and Electronics Engineers (IEEE).

Hiroyuki Kido received B.S. and M.E. in Mathmatics from Waseda University, Tokyo, Japan in 1995, in Information Science from Nara Institute of Science and Technology(NAIST), Nara, Japan in 2002, respectively. Since April 1995, he was at R&D department in JUSTSYTEM CoLtd. and worked for researching Computational Linguistics and developing Japanese Word Processor. Since April 2002, he has been a researcher of Nara Institute of Science and Technology and corporated with JPCERT/CC (Japan Computer Emergency Response Team Coordination Center). His research interests include Internet security architecture and Internet streaming technology.

Katsuyoshi Iida received the B.E., M.E. and Ph.D. degrees in Computer Science and Systems Engineering from Kyushu Institute of Technology (KIT), Iizuka, Japan in 1996, in Information Science from Nara Institute of Science and Technology (NAIST), Ikoma, Japan in 1998, and Computer Science and Systems Engineering from KIT in 2001, respectively. Since Oct. 2000, he has been an Assistant Professor in the Graduate School of Information Science, NAIST. Beginning in Sept. 2001 and ending in Aug. 2002, he spent seven months in total at the Department of Information and Computer Science, University of California, Irvine, on leave of absence from NAIST. His research interests include performance evaluation of networking systems, mobile networks and Internet telephony. He is a member of the WIDE project in Japan, and the Institute of Electrical and Electronics Engineers (IEEE). In 2003, he received the 18th TELECOM System Technology Award, the Telecommunications Advancement Foundation, Japan.

Youki Kadobayashi received the B.E., M.S., and Ph.D. degrees in computer science from Osaka University, Osaka, Japan, in 1992, 1994, and 1997, respectively. He joined Osaka University Computation Center in 1996. From 1999 to 2000, he was an Associate Professor at the Computation Center. He has also been working with the WIDE project as a Board Member since 1998. Since 2000, he has been an Associate Professor of Nara Institute of Science and Technology. His research interests include internetworking architecture and Internet technology for science.

Suguru Yamaguchi received the M.E. and D.E. degrees in computer science from Osaka Unviersity, Osaka, Japan, in 1988 and 1991, respectively. From 1990 to 1992 he was an Assistant Professor in Education Center for Information Processing, Osaka University. From 1992 to 1993, he was with Information Technology Center, Nara Institute of Science and Technology, Nara, Japan, as an Associate Professor. From 1993 to 2000, he was with Graduate School of Information Science, Nara Institute of Science nad Technology, Nara, Japan, as an Associate Professor. Currently, he is a Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. He has been also a member of WIDE Project, since its creation in 1988, where he has been conducting research on network security system for wide area distirbuted computing environment. His research interests include technologies for information sharing, multimedia communication over high speed communication channels, network security and network management for the Internet.