Vol. 40 No. 1 Transactions of Information Processing Society of Japan Jan. 1999

Regular Paper

PLASMA: Multicast/QoS Routing on High Performance Label-switch Networks

Kenji Fujikawa,[†] Masataka Ohta[††] and Katsuo Ikeda[†]

This paper proposes a signaling protocol, *PLASMA*, that provides a mechanism for creating point-to-multipoint channels for multicasting and Quality-of-Service (QoS) guarantee over label-switch networks such as ATM. Senders and receivers specify just a multicast address with or without a QoS request in order to establish point-to-multipoint channels. Multicast channels are soft-state, and are refreshed periodically to dynamically adapt to changes in QoS requests according to the multicast/QoS routing mechanism provided by PLASMA. Furthermore, switches need neither addresses nor identifiers in this mechanism. Internetworking over PLASMA is easily attained by means of the multicasting mechanism, and PLASMA implements all types of communication such as unicasting, multicasting, best-effort, and QoS-guaranteed.

1. Introduction

Quality-of-Service (QoS) guarantee is a hot topic in internetworking. Guaranteeing the QoS for items such as bandwidth and delay makes it practical to transfer multimedia data including video/audio streams. Resource ReSerVation Protocol (RSVP) has been proposed as a resource reservation setup protocol for this purpose.

The QoS controlling function of ATM has not been effectively utilized with respect to the Internet, though ATM was expected to be used as a datalink layer technology for guaranteeing QoS. ATM is now successfully employed only for building IP backbones, and has not gained popularity in the LAN market. The reasons for this are that it is costly, forces complicated settings, and often suffers from low data transfer performance as the result of the loss of even a single cell.

Circuit switches have a mechanism that uses labels, such as ATM's VPI/VCI, placed in frames in order to switch them. This mechanism is suitable for strictly guaranteeing QoS, since it can completely discriminate data flows. From this point of view, ATM is superior to Ethernet or FDDI, which cannot provide strict QoS.

For this reason, we mainly discuss datalink layers that can distinguish data flows according to their labels, and call them "label-switch" networks. We discuss a method for simultaneously implementing multicast routing and QoS routing over label-switch networks. The multicasting mechanism facilitates the construction of label-switch networks such as Ethernets. Our method can also handle unicasting and broadcasting as special cases of multicasting. Consequently, internetworking with multicasting and QoS guarantee functions is easily attained on label-switch networks.

Section 2 describes the features of label-switch architecture and its usage. Section 3 proposes a multicast signaling method called "PLASMA". Section 4 explains the mechanism of multicast/QoS routing. Section 5 evaluates the traffic caused by the signaling messages. Section 6 compares PLASMA with other methods.

2. Label-Switch Architecture

2.1 Definition of Label-switch Architecture

Let us suppose that a datalink layer has the following functions:

- A label is placed in a data frame.
- A sending host determines a label and sends a data frame with the label.
- A switch receives a data frame from one of its point-to-point links, looks up the destination link(s) in its internal table, and transfers the data frame to the destination link(s).
- A receiving host receives the expected data frames, discriminating them according to their labels.

We call a structure with these functions a "label-switch" architecture, and a label placed

---

[†] Graduate School of Informatics, Kyoto University

[††] Computer Center, Tokyo Institute of Technology

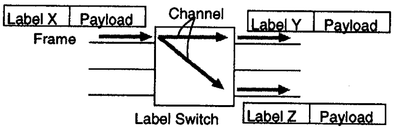


Fig. 1 Label switch architecture.

in a frame a "Layer 2 (L2) label". Each data flow path is set up on each switch in the network before data are actually transferred. This mechanism is called signaling. Therefore, the network easily controls the QoS on each data flow path. We refer to a data flow path as a "channel" in the label-switch network (**Fig. 1**).

ATM and Frame Relay are types of label-switch architecture from this point of view.

2.2 QoS Guarantee for Each Flow

A circuit switch has a mechanism for switching frames according to their labels. This mechanism is suitable for strictly guaranteeing QoS. In addition, it can specify the QoS for each flow even if some flows have the same sender and receiver, since it can assign a unique label to each flow. An IEEE802 LAN such as Ethernet or FDDI cannot assign a path to each flow even if it employs switches, because it only looks destination addresses. ATM is superior to both Ethernet and FDDI, which cannot provide strict QoS at this point (even though they can assign priorities to frames according to IEEE802.1p $^{3)}$).

Simple IP routers, which switch frames (packets) solely on the basis of L3 information such as destination addresses, have difficulty in controlling the QoS for each flow. However, by using port numbers, which are L4 information, or flow labels in IPv6, routers are able to guarantee the QoS for each flow by distinguishing flows. Thus, there is no difference between a router and a label-switch as regards their ability to guarantee QoS. The difference between them is that a router can rewrite neither a destination address, a destination port nor a flow label, while a label-switch can rewrite labels on the path. That is, a router has to allocate global identifiers to data flow paths, while a label-switch may allocate local labels determined by the neighbor switch and itself in order to classify flows. This simplifies the mechanism of label allocation in the label-switch.

2.3 Structure of Networks Using Label-switches

Considering the adaptation of label-switches to real huge-scale networks, we cannot ignore the Internet, which is the only huge-scale data communication network in practice. Thus, we take account of the IP protocol as an L3.

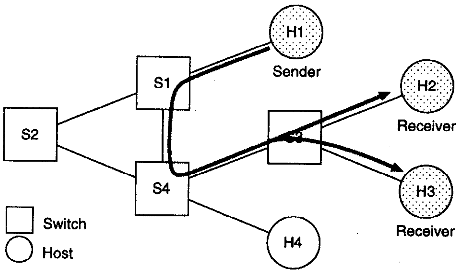


Fig. 2 Basic point-to-multipoint channel.

The Internet comprises many subnets based on various datalink layers, resulting in a huge network, where the subnets are connected by routers. It provides several types of communication, such as unicasting, broadcasting, multicasting, best-effort, and QoS. These types of communication have to be taken into consideration in constructing networks based on label-switches.

Note that the multicasting mentioned in this paper means that a mechanism sends data only by specifying a multicast address, so neither senders nor receivers need to know about receivers or senders. Such a type of multicasting is important not only for broadcasting applications such as remote conferences, but also for autoconfiguration of subnets, where ARP, Neighbor Discovery in IPv6, and/or DHCP are employed.

In order to transport data, label-switches create a point-to-multipoint channel like that shown in **Fig. 2** from a sender to receivers. Basically, that is all that needs to be done. Unidirectional unicasting is regarded as a special case in which there is only one receiver, and bidirectional unicasting is accomplished by using round-trip unidirectional unicasting channels.

A huge data communication network cannot be built solely by using label-switches, since there is a scalability problem. Whenever a flow emerges, all the en-route switches have to set up labels for the flow, even if the data are carried through the path only once. In cases where the switches are used in the backbone, this is impractical with regard to the number of bits for labels or the table size. Label-switches can aggregate flows directed to the same destination host or subnet, but cannot segregate the flows, while IP routers have the capability of both aggregating and segregating flows by using subnet masks. Therefore, introducing IP routers solves the scalability problem. However, QoS-guaranteed transmission is beyond this discussion, because the shortage of bandwidth or other resources related to QoS prevents the allocation of flows before label-switches become unable to manage flows.

MultiProtocol Label Switching (MPLS), of which TagSwitch[5)] is an example, has the same scalability problem as described in Ref. 4).

In the light of these considerations, we discuss a method for constructing a multicast-capable subnet using label-switch, that is, L2 signaling that creates point-to-multipoint channels in a small-scale label-switch network. Here, the term "small-scale network" refers to a network that has a small number of nodes (up to about 300), regardless whether it is a LAN, MAN, or WAN. This signaling provides multicasting with point-to-multipoint channels and QoS guarantee per channel. Each QoS-requested flow is assigned an independent QoS-controlled channel. A label-switch subnet that has these functions is applicable to LANs or backbones such as MAN/WANs, and there is no scalability problem. Consequently, a huge-scale network can be built by using routers to connect these subnets.

Note that though there is a signaling method called SVC (UNI, P-NNI) in ATM, SVC does not provide the multicasting mentioned here.

3. PLASMA Networks

We propose a new L2 signaling protocol called Point-to-point Link Assembly for Simple Multiple Access (PLASMA)[2)]. This section gives an overview of PLASMA.

3.1 Overview of PLASMA Networks

PLASMA provides a simple mechanism for multicasting in a small label-switch network. A sender does not need to know which hosts are receivers, nor does a receiver need to know which hosts are senders there. PLASMA can also specify the QoS for each flow.

Channels are created as a result of L2 label switching at the en-route nodes. Each node employs the "PLASMA protocol," which advertises L2 label switching information in a small network.

A PLASMA node does not have to be assigned its own identifier or address to process the PLASMA protocol, unless it is a terminal node in terms of data transportation on L2. Therefore, PLASMA allows autoconfiguration of subnets.

In a PLASMA network (a label-switch network based on PLASMA), nodes can be connected in any topological manner. Of course, a PLASMA network may contain loops or redundant paths, thus realizing a flexible configuration.

One domain in which the PLASMA protocol is exchanged corresponds to one IP subnet when IP is used as L3. All users have to do to auto-configure an IP subnet is connect PLASMA nodes, turn them on, and assign IP addresses to hosts. Furthermore, users may be freed from IP address assignment using DHCP.

3.2 Functions and Types of Nodes in a PLASMA Network

An entity that processes the PLASMA protocol is called a "node" in PLASMA. A node executes the following functions:

- data sending/receiving
- L2 switching
- L3 forwarding (e.g., IP packet forwarding)

Nodes are composed by combining these functions, and are categorized into the following types:

- host (sending/receiving data)
- switch (L2 switching)
- router (sending/receiving data, L3 forwarding)
- switching router (sending/receiving data, L2 switching, L3 forwarding)

A PLASMA network is configured by connecting these nodes.

3.3 Messages of PLASMA Protocol

The PLASMA protocol uses mainly two types of messages, a "NOTIFY message" and an "ACCEPT message", in order to exchange the label information and create channels. A NOTIFY message transmits information on the channels that a sender wants to create from the sender to receiver(s) by flooding. An ACCEPT message is sent from the receiver(s) to the sender along the reverse path of the channel to be actually established, and transmits the L2-label information that identifies the channel. These messages are transmitted through the well-known channel for signaling by the adjacent nodes, and forwarded hop-by-hop.

PLASMA channels are distinguished uniquely by pairs of source address and channel identifier placed in NOTIFY or ACCEPT messages. That is, when two channels have the same pair, they are regarded as the same channel.

Nodes must send PLASMA protocol messages periodically to maintain the channels. A channel expires if after a defined period of time nodes have not received related PLASMA protocol messages. Therefore, channels in PLASMA are “soft-state”.

4. Multicast/QoS Routing in a PLASMA Network

This section explains how multicast/QoS routing works in a PLASMA network. The QoS for each data flow can be preserved by assigning a channel to each flow and setting the QoS parameters such as the bandwidth on the channel.

4.1 QoS Routing

The purpose of QoS routing is to find the best or next best path that satisfies the specified QoS. The current routing in the Internet is based solely on the delay (hop count), which is one of the parameters of QoS. This section considers QoS routing that first obtains the bandwidth and then minimizes the delay.

Multicasting on the Internet permits multiple senders and receivers to join a multicast group. In addition, every receiver can dynamically control the QoS by using RSVP. In other words, a receiver of multicasting can start and stop specifying the QoS and can change the QoS as desired. This also means that the receiver may request best-effort access to the data. In order to adapt to such a flexible QoS, the network must be able to change the structure of multicast trees according to either or both of new join requests and/or QoS change requests from receivers when employing source-routed trees as multicast trees. In the case of shared trees, new join requests from senders have to be considered too. The method named “route pinning[7]”, by which a sender or a receiver determines the route of transmission in advance, is not suitable for QoS routing, since it cannot dynamically change routes.

A PLASMA network uses source routed trees as multicast trees, and provides the QoS routing that allows receivers to join and leave dynamically. PLASMA needs to assign neither addresses nor identifiers to switches in the case of QoS routing.

4.2 NOTIFY and ACCEPT Messages for QoS Routing

For QoS routing, NOTIFY and ACCEPT messages are denoted as $N(S, C, D, B_0 : H_0, B_1 : H_1, \ldots)$ and $A(S, C, L, B)$, respectively, where:

- S: source address
- C: channel identifier
- D: destination address
- B_n: bandwidth of nth path
- H_n: hop count of nth path
- L: L2 label
- B: bandwidth actually used on channel

A NOTIFY message conveys one or more pairs of bandwidth and hop count. However, it does not necessarily convey all the pairs found by flooding. Invalid paths are discarded when NOTIFY messages are merged at nodes. A path is valid provided that the NOTIFY message does not have any other paths whose bandwidth is larger or equal and whose hop count is smaller or equal. For example, against a path with 5 Mbps bandwidth and 3 hops, a path with 2 Mbps and 2 hops is valid, while a path with 2 Mbps and 5 hops is not.

A NOTIFY message for QoS contains information on just one channel in it, since channels may be different even if their destinations are the same, while a NOTIFY message for best-effort can contain information on one or more channels.

4.3 Example of QoS Channel Creation

We explain how to create channels on an example network in Figs. 3 to 9.

The PLASMA network in **Fig. 3** is configured with five hosts H1, H2 ... H5 and six switches S1, S2 ... S6 The bandwidth of the link from switch S1 to S5 is 5 Mbps in this direction, and that of the link from S5 to S6 1 Mbps. We assume in the following discussion that there is plenty of bandwidth for other links.

In Fig. 3, host H1 is going to create a channel to address M1 with a maximum bandwidth of 5 Mbps, and NOTIFY messages are shown in this situation[☆]. In **Fig. 4**, H2 simultaneously requests a channel to M2 with a maximum bandwidth of 2 Mbps, and NOTIFY messages are shown. H5 is notified that there are two paths with different bandwidths.

When host H4 requests a channel to address M1 with 5 Mbps, an ACCEPT message is sent as shown in **Fig. 5**. As a result, a channel is established, and the bandwidth of the link from switch S1 to S5 is reduced to 1 Mbps (note that this has nothing to do with the bandwidth in the opposite direction).

In this situation, unless switch S1 is aware

[☆] The reason a NOTIFY message does not go to some nodes – for instance, from S1 to S2 – is that JOIN messages that suppress redundant NOTIFY messages are employed[2].

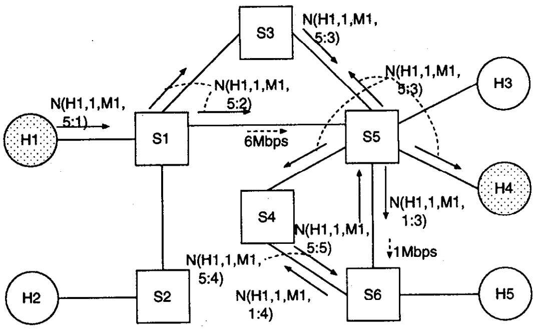


Fig. 3 NOTIFY message from H1 to M1.

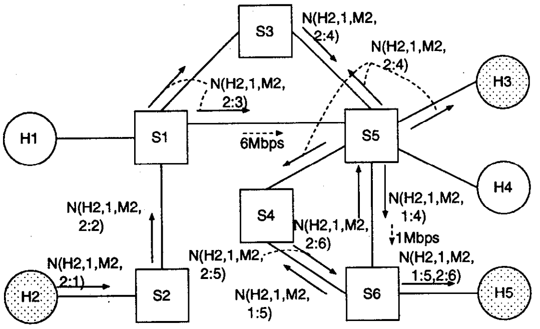


Fig. 4 NOTIFY message from H2 to M2.

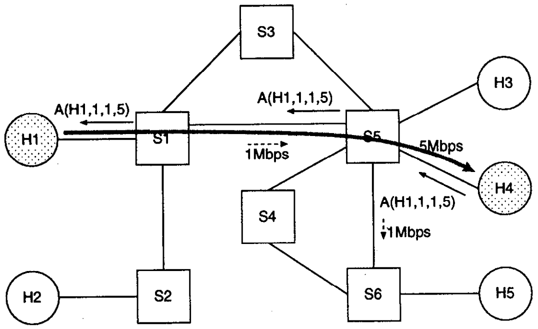


Fig. 5 Channel from H1 to M1 with 5 Mbps.

that the channel created by host H1 reduces the bandwidth of the link from S1 to S5, it moves the channel to the link from S1 to S3 with sufficient bandwidth, judging that the link from S1 to S5 is no longer available for the channel, since that link now holds a bandwidth of 1 Mbps. This problem is known as the route-flapping problem. However, continuous route flapping does not occur in PLASMA, because NOTIFY messages provide switches with correct information about bandwidth☆. Thus, S1

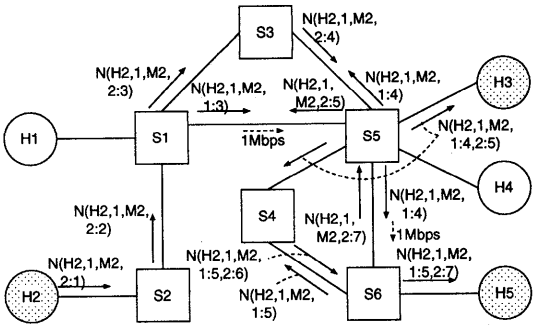


Fig. 6 Changed NOTIFY message from H2 to M2.

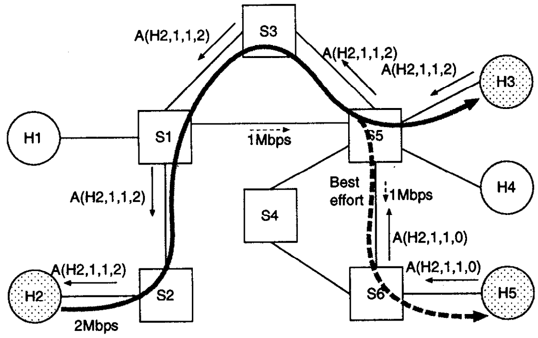


Fig. 7 Channel from H2 to M2 with 2 Mbps/best-effort.

comes to know that the channel from H1 to M1 consumes 5 Mbps, even though the link from S1 to S5 is 1 Mbps. Consequently, S1 does not change the path of the channel.

The channel creation in Fig. 5 causes the NOTIFY message from host H2 to address M2 to change into that depicted in **Fig. 6**, since the bandwidth of the link from switch S1 to S5 is changed to 1 Mbps.

Then, when host H3 requests a channel from H2 to address M2 with 2 Mbps, and H5 requests with best-effort bandwidth, the ACCEPT messages shown in **Fig. 7** are sent from H3 and H4. These messages are delivered by selecting a path that can afford sufficient bandwidth, which has been made known by the NOTIFY messages in Fig. 6 on the reverse path of the NOTIFY messages. Switch S5 receives the two ACCEPT messages with different requested bandwidths, merges them, and forwards an ACCEPT message with higher bandwidth upstream. Consequently, the channel in Fig. 7 is created.

Assuming that the channel from host H1 to address M1 is deleted, the NOTIFY messages will change back into the one shown in Fig. 4. Therefore, the ACCEPT messages and channel

---

☆ Though a delay of flooding NOTIFY messages sometimes causes route flapping, the route flapping immediately converges, since a NOTIFY message with a smaller hop has precedence.

Table 1 Number of messages per switch.

	BE uni	QoS uni	BE multi	QoS multi
NOTIFY	n	nf	n	nf
ACCEPT	pnf/sl		$\min(rp, L+r+1)nf/sl$	

BE: Best Effort, uni: unicast, multi: multicast

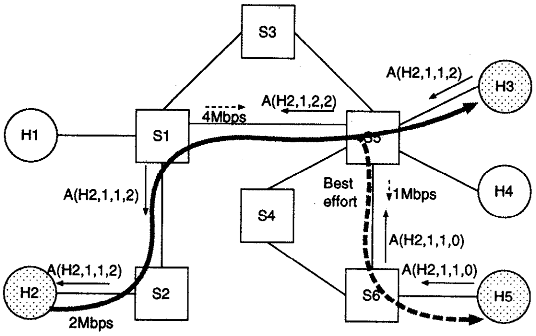


Fig. 8 Changed channel from H2 to M2 with 2 Mbps/best-effort.

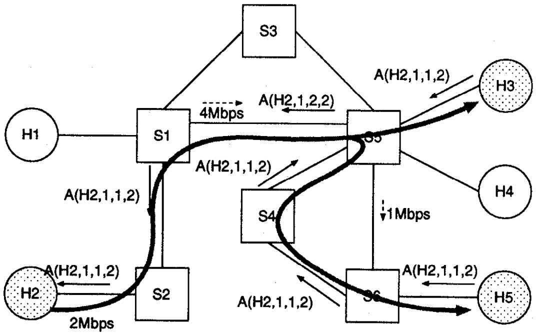


Fig. 9 Changed channel from H2 to M2 with 2 Mbps.

will be modified as shown in **Fig. 8**.

Furthermore, if host H4 alters the requested bandwidth from best-effort to 2 Mbps, then ACCEPT messages and the channel will be changed as shown in **Fig. 9**.

4.4 Considerations on QoS Route Change

In QoS routing on PLASMA, a path for a certain receiver may change provided that at least one of the following conditions is met:

- A switch is added or removed.
- A new path with a smaller hop count becomes available.
- The receiver changes its QoS request.
- Another receiver for the same channel changes its QoS request into a larger one.

These are inevitable route changes for QoS routing with multicasting, but they do not result in route flapping. For example, in Fig. 9, even if the NOTIFY message from H1 to M1 is projected again, the channel from H2 to M2 will not be modified this time.

Thus, the PLASMA protocol supplies a multicasting mechanism capable of both controlling the QoS and stabilizing routes.

5. Traffic for Signaling Messages

The traffic for PLASMA signaling messages may become a problem. The number of signaling messages tends to increase in exchange for the advantage that switches require neither addresses nor identifiers. We evaluate the signaling message traffic in PLASMA networks, and then show that it does not load the networks.

In a PLASMA network, let us use the following symbols:

- n: number of hosts
- s: number of switches
- l: number of links per switch
- L: total number of links that connect switches in the whole network
- f: number of flows per host
- p: hop count of the longest loopless path between two hosts
- r: number of receivers of multicasting per flow

The upper bound on the number of messages one switch sends or receives per link can then be estimated as shown in **Table 1**.

We explain the number of ACCEPT messages for multicasting shown in Table 1. The number of ACCEPT messages for a channel is equal to the number of links used to create the channel. Thus, assuming that for a certain multicast channel the length of each path from the sender to each receiver is p, then the number of links for the channel, that is, the number of ACCEPT messages, is rp. However, it cannot exceed the sum of the total number of links, L, and the number of links towards the receivers and the sender, $r+1$. Thus, the number of ACCEPT messages is $\min(rp, L+r+1)$, and the upper bound per switch's link in the whole network is $\min(rp, L+r+1)nf/sl$.

We estimate the traffic caused by PLASMA signaling messages in the network shown in **Fig. 10** according to Table 1. In Fig. 10 (a), eight hosts are connected to a level-one switch.

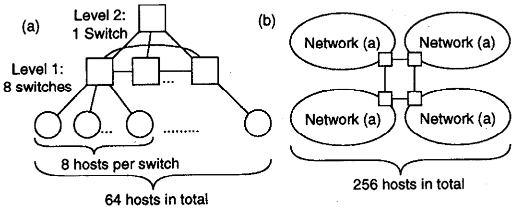
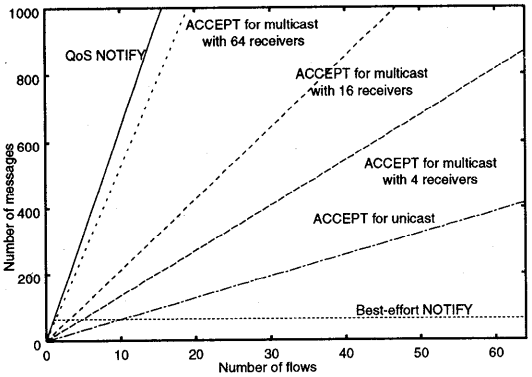
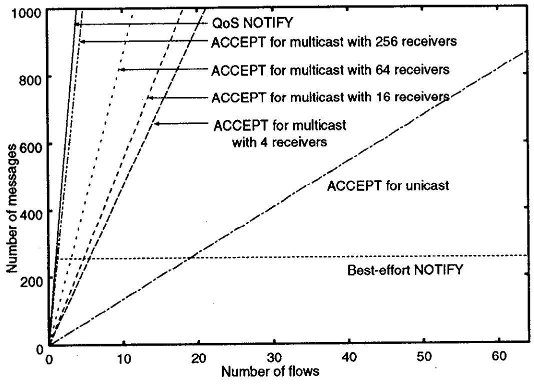


Fig. 10 Simulation network.



(a) 64 hosts ($n = 64$, $s = 9$, $l = 11$, $L = 16$, $p = 10$)



(b) 256 hosts ($n = 256$, $s = 36$, $l = 11$, $L = 68$, $p = 21$)

Fig. 11 Number of messages per link.

Eight level-one switches are connected to two other level-one switches, and also connected to the only level-two switch. Thus, this network comprises 64 hosts and nine switches. In Fig. 10 (b), four type-(a) networks are connected with a loop, and there are 256 hosts and 36 switches.

Figure 11 shows the number of messages per switch's link over the network shown in Fig. 10, when the number of flows per host varies from zero to 64.

The graphs show that the QoS NOTIFY message is the most significant issue. For example, assuming that a switch can process up to 1,000 messages per link, one host can create 15.6 flows on the average in the case of 64 hosts, and 3.9 flows in the case of 256 hosts. This is practical enough in a LAN environment. Then, assuming that the size of a message is 100 bytes, that a node sends a NOTIFY or ACCEPT message for a certain flow every 10 seconds, and that a switch sends or receives 1,000 messages per link, the switch sends or receives 100 messages/sec, which yields a traffic level of 78 kbps per link. The traffic created by the signaling messages does not become a problem at all, since PLASMA targets high-performance networks with bandwidths of 100 Mbps or more.

Consequently, PLASMA creates sufficient flows in practical use, and the traffic caused by the signaling messages is trivial in a high-performance label-switch network.

6. Comparison with Other Methods

6.1 Comparison of Signaling Methods

ATM switches that employ P-NNI 1) are hierarchical, and thus require settings, in the same way as IP routers. This is inevitable in view of the fact that P-NNI is designed to handle a huge-scale network by itself. However, since a huge network consisting solely of ATM is unlikely to appear, we should take account of the need to use ATM for the Internet. In this case, hierarchical ATM signaling is not necessarily inevitable. Furthermore, P-NNI does not have an (IP-like) multicast mechanism, but only a QoS routing mechanism based on route pinning, which is also not suited to multicasting.

On the other hand, PLASMA switches require no settings and can support multicasting/QoS routing.

6.2 Comparison of Methods for Managing Bandwidth

IEEE 802 does not have a method of checking the remaining bandwidth in a network. IEEE802.1p enables a network to assign priorities to frames, but even in this case, the network is unable to know the correct bandwidth. In order to obtain information on the bandwidth, a Subnet Bandwidth Manager (SBM) 6) has been proposed that is laid and manages the bandwidth in an IEEE802 LAN.

PLASMA can accurately obtain information on the available bandwidth of each path from a certain end host to another by means of NOTIFY messages. For this mechanism, it does not require a server like SBM. Bandwidth information is managed locally and independently.

7. Conclusions

We have proposed a signaling protocol named PLASMA that provides a mechanism for creating point-to-multipoint channels using multicast/QoS routing over label-switch networks such as ATM. Senders and receivers specify a multicast address either with or without a QoS request in order to establish point-to-multipoint channels. Multicast channels are soft-state and refreshed periodically to dynamically adapt to changes of QoS requests in PLASMA. In addition, switches need neither addresses nor identifiers for this mechanism.

We evaluated the traffic of the PLASMA signaling messages, and pointed out that it does not become a problem.

We have already implemented the PLASMA protocol, and constructed an IP/PLASMA LAN and WAN using ATM. Internetworking over PLASMA is easily attained by means of the multicasting mechanism, and implements all types of communication such as unicasting, multicasting, best-effort, and QoS-guaranteed.

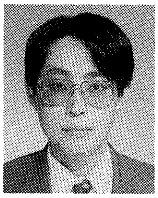
As further tasks, we are going to adapt IPv6 to PLASMA with regard to an L3, and PLASMA to Fast/GigabitEthernet with regard to an L2 for the future works. Even in such datalink layers, PLASMA can be applied simply by embedding labels in data frames. This could be accomplished by making use of local MAC addresses as labels.

Acknowledgments We thank members of Internet Technical Research Committee (ITRC) for helping us in many ways.

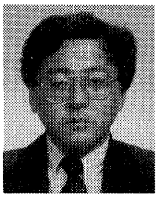
References

1) ATM-Forum: P-NNI Specification Version 1.0 (1996).
2) Fujikawa, K., Ohta, M. and Ikeda, K.: IP/ATM Networks with Straightforward Multicasting and QoS Assurance, *INET '97* (1997).
3) IEEE: Supplement to MAC Bridges: Traffic Class Expediting and Dynamic Multicast Filtering, *IEEE802.1p* (1997).
4) Ohta, M.: Hop, Step, but Don't Jump for Scalable Lower Layer Forwarding of Best Effort Traffic, *INET '97* (1997).
5) Rekhter, Y., Davie, B., Katz, D., Rosen, E. and Swallow, G.: Cisco Systems' Tag Switching Architecture Overview, RFC2105 (1997).
6) Yavatkar, R., Hoffman, D., Bernet, Y. and Baker, F.: SBM (Subnet Bandwidth Manager): A Proposal for Admission Control over IEEE 802-Style Networks, Internet draft (work in progress) (1997).
7) Zappala, D., Estrin, D. and Shenker, S.: Alternate Path Routing and Pinning for Interdomain Multicast Routing, USC Computer Science Technical Report (1997).

(Received May 1, 1998)
(Accepted November 9, 1998)



Kenji Fujikawa was born in 1970. Received BE and ME in the field of Information Science from Kyoto University in 1993 and 1995, respectively. From 1997 he was a research associate of department of Information Science and from 1998 is a research associate of Graduate School of Informatics in Kyoto University. His research interest is high-speed internetworking. He is a member of IPSJ and ISOC.



Masataka Ohta was born in 1959. Received BSC, MSC and DSC in the field of Information Science from University of Tokyo in 1982 and 1984 and from Tokyo Institute of Technology in 1994, respectively. From 1987 he is a research associate of department of Computer Center in Tokyo Institute of Technology. His research interests include computer graphics, high-performance computing, character code and internetworking. He is a representative of ADSL research group, a chair of APNG internationalization working group, a secretary general of ISOC Japan chapter and a member of IPSJ and ACM.

Katuso Ikeda was born in 1937. Received BE and ME in the field of Electronic Engineering, and DE in the field of Computer Science from Kyoto University in 1960, 1962 and 1978, respectively. From 1978 he was a professor of Institute of Information Science and Electronics in University Tsukuba. From 1988 he was a professor of Department of Information Science and from 1988 is a professor of Graduate School of Informatics in Kyoto University. His research interest is centered around construction of Intelligent Information Media, which assists human daily activities. He is a member of IPSJ, JSAI, IEEE and ACM.