Vol. 46 No. 2 IPSJ Journal Feb. 2005

Regular Paper

Network for Video Distribution between Preceding and Succeeding Vehicles on the Same Route

KATSUHIKO SATO,[†,††] MICHIAKI KATSUMOTO[†††] and TETSUYA MIKI[††]

The Future Vision Distribution Service (FVDS) (Sato, et al., 2004) is an innovative video delivery service concept based on the Internet ITS (Intelligent Transport System) area, and Source Mobility Support Multicasting (SMM) (Sato, et al., 2004) consists of new multicasting techniques needed to realize FVDS. However, FVDS and SMM lack sufficient application-level protocols to manage preceding and following vehicles on the same route, and have a drawback of topological constraints in the access network to accommodate micro-mobility. To solve these problems, this paper presents two new protocols. The Mobile Vehicle Management (MVM) protocol is an application-level protocol that discovers a preceding vehicle on the same route and provides its multicast address to following vehicles by using information from the global positioning system (GPS) and a route-guiding function. The Self-Organizing Tree (SOT) protocol is a network-level protocol for use in the access network that organizes radio base-stations into a logical tree topology in a self-forming and self-healing manner. This eliminates the topological constraints and provides the network with robustness against failures and flexibility for network design. To show how these protocols further the implementation of FVDS and SMM, this paper also gives details of a software implementation model for each network node system; specifically, we have designed a software architecture, composition elements for each protocol and their relations, and internal state-machines for each element.

1. Introduction

A great deal of research and development has gone into intelligent transport systems (ITSs) based on Internet technologies[3)~7)]. An ITS is generally aimed at providing vehicles and road infrastructure with intelligent functionality to alleviate serious traffic congestion and traffic accidents. Applying the latest Internet technologies, such as broadband, mobile, and multicast communication techniques, to an ITS will greatly enhance the system's ability to provide innovative information distribution services to any user on the road. In an example of advanced Internet ITS services, a real-time video distribution service has been proposed, where surveillance video cameras deployed at construction sites or congestion points provide video information for drivers and other traffic control systems through wireless meshed network technologies[4)]. In addition, a system has been developed where each vehicle is equipped with a video camera so that it can distribute video information to other vehicles in the vicinity via multi-hop routing and multicasting techniques[5)].

The Future Vision Distribution Service (FVDS)[1)] proposes a more advanced concept based on ITS service. In this concept, earlier vehicles and later vehicles on the same route are identified by using the global positioning system (GPS) and a route- guiding function (i.e., a car-navigation system), and a preceding vehicle multicasts real-time video information to vehicles behind it; we call this advance information "future vision." The Source Mobility Support Multicasting (SMM) technique[2)] provides a network distribution functionality (protocol) to realize the above concept. It guarantees consecutive video distribution during high-speed movement of both the multicast source and receivers by using the Cellular IP infrastructure[14)~16)].

However, two major problems remain to be solved in FVDS and SMM. First, for FVDS we need a means of managing the preceding and following users on the same route; for example, we need to know how to discover an appropriate preceding user and provide his or her multicast address to the users behind. Second, because it uses the Cellular IP infrastructure, SMM requires an access network to form a physical tree topology. A tree topology is in general vulnerable to network failures and inflexible for covering any road topology. Such a topological constraint should be eliminated to enable im-

---

† Japan Radio Co., Ltd
†† The University of Electro-Communications
††† National Institute of Information and Communications Technology

plementation of the proposed scheme.

Focusing on these points, this paper proposes two protocols. The Mobile Vehicle Management (MVM) protocol, an application-level protocol between the client system on the vehicle and the server system in the network, solves the first problem. The server manages the current position and traveling route of each client's vehicle by using information obtained from a car-navigation system, and provides a later client's vehicle with a preceding client vehicle's multicast address. The self-organizing tree (SOT) protocol is a network-level protocol used in the access network to solve the second problem. It allows radio base-stations to organize a logical tree topology in a self-forming and self-healing manner by using a spanning tree algorithm[23)], which provides the network with robustness and flexibility. To further the implementation of FVDS and SMM, we also describe a software implementation model at each network node system. More specifically, we depict the software architecture, composition elements for each protocol and their relations, and internal state-machines for each element.

The remainder of this paper is organized as follows. Section 2 explains our previous work, the FVDS concept, and the SMM protocols. Section 3 describes the MVM protocol. Before describing the protocol, we reconsider the current SMM protocol architecture and show how it can be improved. Section 4 explains the SOT protocol. Section 5 describes our software implementation model.

2. Previous Work on the Future Vision Distribution Service

2.1 Concept of the Future Vision Distribution Service

FVDS groups vehicles going to the same destination or in the same direction, and earlier vehicles on the same route provide real-time video information to subsequent vehicles via a real-time streaming technique. For the later vehicles, this video information provides future vision, i.e., a chance to look at conditions ahead of them.

Vehicles exchange information regarding their route to a destination and location (running point), which can be obtained through the global positioning system (GPS) and a route-guiding function, with other vehicles in the same group through the network. Each vehicle finds a preceding vehicle and receives real-time video information from it. That is, a preceding vehicle provides later vehicles with a camera image of what it is currently seeing. In the example shown in **Fig. 1**, vehicle A is traveling at a point ahead of vehicles B and C. Vehicle A multicasts its camera data to B and C. Vehicle C is also at a point ahead of vehicle D and E, and multicasts its camera data to D and E.

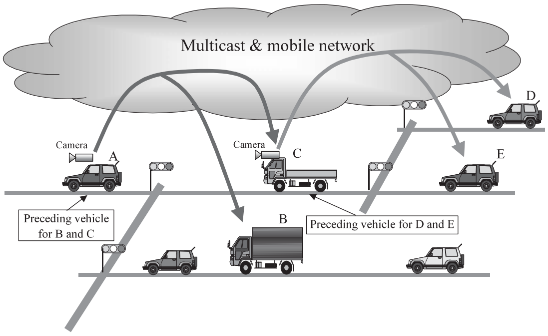

Fig. 1 Future-vision distribution service.

This service will be useful to ITS users because it enables drivers to see what lies ahead along their route (e.g., weather or traffic conditions) by accessing visual information provided by the drivers who have preceded them along that route.

2.2 Source Mobility Support Multicast (SMM)

The FVDS requires a special network scheme that enables high-speed mobile hosts to distribute high-quality video streaming to multiple points without any service disruption. The source mobility support multicast (SMM) technique[2)] has been developed to ensure such seamless multicast distribution even in high-speed mobile environments.

Before SMM, much of the research done on multicasting to support host mobility was based on the mobile multicast (MoM) technique[17)~21)], which uses Mobile IP techniques[11)~13)]. Fundamentally, though, Mobile IP cannot eliminate the route redundancy problem, nor can it enable consecutive communication such as through use of a diversity handover mechanism. Therefore, SMM uses techniques of Cellular IP[14)~16)], which is a representative IP-based network scheme supporting micro-mobility functionality. In Cellular IP, switch nodes have routing caches to forward data packets, through which a data-forwarding path is dynamically established every time data packets originate from a mobile host. Some switch nodes have paging caches to manage the location of the host through which the Page message is broadcast in a paging area to page a mobile host. One important feature of Cellular IP is its support of a diversity handover mechanism whereby multiple data-forwarding paths are established simultaneously while a mobile host is moving between two radio base stations.

SMM gains an added multicast forwarding functionality on Cellular IP by putting a multicast forwarding cache entry (MFCE) on switch nodes. The SMM protocol constructs both a source point tree and a rendezvous point tree in each multicast tree, which minimizes the overhead in reconstructing multicast trees (which is especially likely to be caused by a multicast source movement). The SMM architecture and the details of its protocols are described in Ref. 2), which also describes ways to enable wide SMM operability by partitioning the network and confining protocol uniqueness and some topological constraints within the local access networks.

3. Application-Level Procedures: Mobile Vehicle Management

This section presents FVDS application-level procedures for route registration and preceding-vehicle discovery functions. Before describing the MVM protocol, we reconsider the current SMM protocol architecture described in Section 3.1 of Ref. 2) and show how it can be improved. On the basis of the new protocol architecture, we describe MVM as application-level procedures and behaviors of the underlying network-layer protocol, which are modified from those in Ref. 2).

3.1 Reconsideration of Network Architecture and Protocol Deployment

To provide operational scalability, as shown in **Fig. 2**, a network hierarchically consists of a single core network and multiple access networks with gateways (GWs) located at their boundaries. In the core network, both unicast and multicast data packets are switched through layer-3 functionalities. To allow the network to form any topology, standard protocols such as the Open Shortest Path First (OSPF)[8] and Protocol Independent Multicast-Sparse Mode (PIM-SM)[9] are implemented. As mentioned in Ref. 2), each GW acts as a source point of a multicast tree, and source point trees (SPTs) are constructed for each multicast group. A bootstrap router (BSR) collects the source point information and floods the list of source points through a bootstrap message. In the access network, all network nodes run the Cellular IP protocol. They organize as a tree topology, whose root is a GW and whose leaves are radio base-stations. The network is divided into several paging areas. Servers (SVs) at the application level are deployed at a GW for each access network. These servers manage each vehicle's current position and traveling route, search for an earlier vehicle on the same route, and provide a multicast address to a requesting vehicle. An upper-level server (USV), which has a list of SV addresses corresponding to geographical addresses for the access network of the SV, is deployed in the core network.

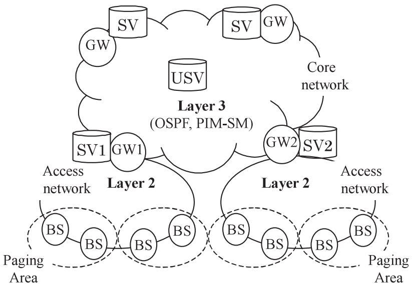

Fig. 2 Network architecture.

Here, we reconsider how the Cellular IP protocol operates in the access network. As is generally known, the IP routing scheme is basically a form of proactive routing where every network node (i.e., router) and attached network or host is uniquely assigned a permanent IP address and every router calculates in advance the direction in which the destination address of a received packet is to be forwarded. In contrast, the Cellular IP routing scheme is a form of reactive routing which dynamically learns the direction in which the destination address of a received packet is to be forwarded by receiving a *route update* message from the host. (Note that every network node predetermines its interfaces with respect to an upstream port and some downstream ports according to the tree topology.) The routing scheme determines routes on an on-demand basis and thus does not require any pre-assigned address on a network node and an attached network. Such a routing scheme is similar to a broadcast-based layer-2 network, such as Ethernet, where received packets are basically broadcast, except for those whose destinations are already learned. Therefore, the routing scheme of Cellular IP can use

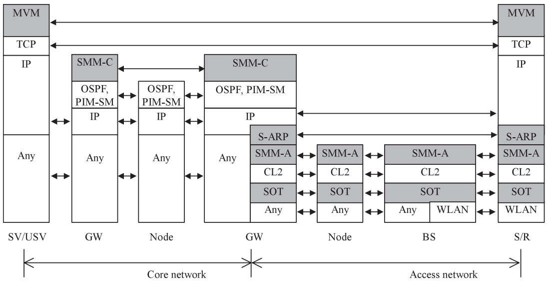

Fig. 3 New protocol stack: SMM-A and Cellular IP (CL2) are located at the layer-2 stack. MVM, SOT, and S-ARP are newly added.

a layer-2 network scheme rather than a layer-3 scheme. In this paper we demonstrate that Cellular IP can be operated as a layer-2 network, where a MAC address instead of an IP address is used for both unicast and multicast routing.

Figure 3 shows a protocol stack. In the access network, switch nodes and radio base-stations run Cellular IP and SMM-A as a layer-2 network (hereafter Cellular IP for layer 2 is expressed as CL2). In addition, they run the SOT protocol with which radio stations organize a logical tree topology in a self-forming manner (as detailed in Section 4). In the core network, SMM-C is operated as an inter-GW protocol over PIM-SM and OSPF. The MVM protocol is run between the SV and the client (we will express the multicast source and receiving members as S and R, respectively) as an application-layer protocol for route registration and preceding-user discovery.

Note that we must define an address resolution protocol (ARP) specialized for this layer-2 access network, since CL2 is a non-broadcast network. The S-ARP (APR for SMM) protocol is thus newly located at the GW acting as an ARP server, as is used in a connection-oriented layer-2 network such as an ATM network $^{22)}$.

3.2 Mobile Vehicle Management Protocol

As shown in **Fig. 4**, any user (who could be a preceding user) sends an SV a *Register* message, which includes the user's multicast address and traveling route from the current position to the destination, $C(g) = \{V_0, V_1, \ldots, V_i\}$, whenever he or she crosses an access network boundary. Here, g is the multicast address of the user, $\{V_0, V_1, \ldots, V_i\}$ is a set of intersections the user will pass through on the way to the destination, and V_i denotes the coordinates of an intersection location (x, y), with the intersection being closer to the destination as i increases.

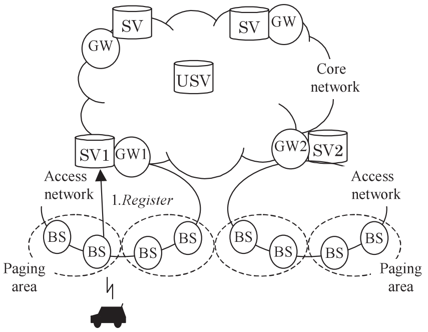

Fig. 4 Application-level procedure 1: a user registers his/her multicast address, current position, and traveling route (list of intersections), and updates his/her current position whenever he/she passes through an intersection.

A user might send, for example, multicast address g = G and route $\{\mathrm{K, A, T, J, Q}, \ldots\}$ to SV1. V_0 = K is the current position of the user. In its database, SV1 has pairs of multicast addresses and route information for all users in the access network it manages. Now $C(\mathrm{G}) = \{\mathrm{K, A, T, J, Q}, \ldots\}$ has been registered in SV1. As long as the user stays within the access network, he or she sends a *Register* message (including only its current position V_0) whenever he or she passes through an intersection. The SV receives the *Register* message and updates the existing $C(g)$. For example, once the user has moved to the next intersection 'A', he or she sends $V_0(=\mathrm{A})$ to SV1 and the existing $C(\mathrm{G})$ in SV1 is updated to $C(\mathrm{G}) = \{\mathrm{A, T, J, Q}, \ldots\}$.

If the intended route of a user changes before he or she reaches the destination, the user immediately sends a *Register* message which includes a new set of intersections $\{V_0, V_1, \ldots, V_i\}$ to the SV.

Member R requires the future vision at the point where the above user is traveling, as shown in **Fig. 5**. Therefore, R sends a *Request* message to SV2, including its unicast address and its route from the required position to the destination, $R(u) = \{V_0, V_1, \ldots, V_i\}$. Here, u is the unicast address of R which the SV will use to reply with a multicast group address to R. For example, R requests $R(\mathrm{U}) = \{\mathrm{A, T, J, F}, \ldots\}$. $V_0(=\mathrm{A})$ is the position at which R wants to start seeing what lies ahead. SV2 receives the *Request*, discovers that $V_0(=\mathrm{A})$ is outside of its management areas, and so sends *Query* messages including $R(\mathrm{U}) = \{\mathrm{A, T, J, F}, \ldots\}$ to the USV. The USV has a list of SV addresses corresponding to geographical

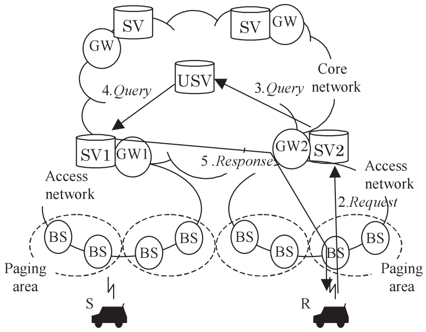

Fig. 5 Application-level procedure 2: a later user R requesting future vision sends the required position and his/her traveling route to SV2. The request is forwarded to SV1, and SV1 looks up an appropriate preceding user, S, and then sends the multicast address of S to R.

Table 1 MVM messages.

Message	Parameters	Direction
Register (New)	User's IP multicast and unicast address, current position and travel route (list of intersections)	Any user -> SV
Register (Update)	IP multicast and unicast address, current position	Any user -> SV
Query	Requesting user's IP unicast address, requested route (list of intersections)	Requesting user -> SV, SV -> USV, USV -> Target SV
Response	Precedent user's IP multicast and unicast address, matched route (list of intersections)	Target SV -> Requesting user
Status	Precedent user's IP multicast and unicast address, current position and next intersection	Precedent user -> Requesting user

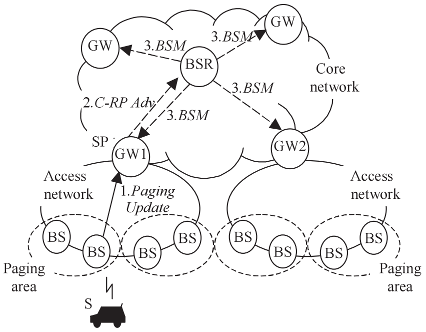

Fig. 6 Network-level procedure 1: whenever a user crosses a paging area, he/she reports his/her multicast group address and paging area ID (obtained from a BS) to GW1. GW1 becomes the source point of multicast tree for the group in the core network, and notifies the BSR of the rendezvous point candidate. The BSR floods this information throughout the core network.

addresses covered by the access network of each SV. Using $V_0 (= \mathrm{A})$ as a search key, the USV searches for a corresponding access network and determines the SV1 to which it should forward the *Query*. The SV1 receiving the *Query* looks up the $C(g)$ database using $\{V_0, V_1, \ldots, V_i\}$ in the *Query* as a search key and obtains only one $C(g)$ with the longest matched route. At least V_0 and V_1 must be matched to those in $C(g)$; that is, the route to the next intersection from the position at which R wants to start seeing ahead must at least be matched. Here, $C(\mathrm{G}) = \{\mathrm{A}, \mathrm{T}, \mathrm{J}, \mathrm{Q}, \ldots\}$ is matched as the longest match (i.e., $i = 2$). The user with multicast address G is then designated as the preceding user; i.e., as multicast source S for R. After that, SV1 sends a *Response* message to R using unicast address $u (= \mathrm{U})$, and this message includes the multicast address G and part of the matched route $H(\mathrm{G}) = \{\mathrm{A}, \mathrm{T}, \mathrm{J}\}$.

Member R receives the *Response*, confirms the received route $H(\mathrm{G}) = \{\mathrm{A}, \mathrm{T}, \mathrm{J}\}$, and joins multicast group G. The network provides multicast data forwarding by constructing a multicast distribution tree. (We explain the network-level procedures in detail in the next section.) While sending multicast data, S multicasts *Status* messages including its current position and next intersection $I(g) = \{V_0, V_1\}$ through the same multicast distribution tree (using the same multicast address G) every time it passes an intersection. R confirms S's route by comparing the received $I(g)$ and $H(g)$ (i.e., R can confirm S is still traveling on the same route as R and maintaining the distance from R that R initially desired). If R finds that S has departed from its initial route, it sends a new *Request* to the SV to find another preceding user.

Table 1 shows the list of messages defined in this protocol.

3.3 Behaviors of Other Network Layer Protocols

As shown in **Fig. 6**, any user (who could be multicast source S) transmits a *Paging Update* message (specified in CL2) to GW1 to register his or her existence (i.e., update the paging cache table) whenever he or she crosses a paging area boundary. The *Paging Update* message includes the user's IP and MAC address for both unicast and multicast and paging area ID obtained from a radio base station. Intermediate nodes between the user and GW1 forward the *Paging Update*. The intermediate nodes managing a paging cache table update their paging cache entry for both the unicast and multicast MAC addresses. The pair consisting of the IP and MAC unicast addresses will be managed on the ARP server located at GW1 if the *Paging Update* is the first one in this access network. Upon receiving the *Paging Update*, GW1 sends a *C-RP Adv* (rendezvous point candidate advertisement as specified in the PIM-SM) to the

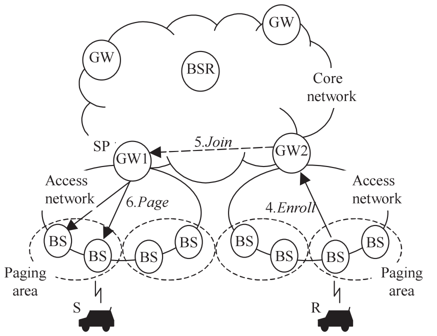

Fig. 7 Network-level procedure 2: a receiver who has obtained the multicast group address of a preceding user (by MVM) sends an *Enroll* message to GW2 to join the multicast group. GW2 sends a PIM *Join* message to GW1. GW1, if it has not received multicast data, sends a *Page* message to urge the source host to send multicast data.

BSR, which then says that GW1 has become the source point (SP) of the multicast tree for the group in the core network. The BSR sends a bootstrap message (*BSM*), which includes the GW1's IP address and the multicast IP address for the group, to all GWs in the core network. Note that these procedures are well known as the PIM-SM bootstrap mechanism.

Member R sends an *Enroll* message (specified in the SMM-A) for the multicast group to GW2 (**Fig. 7**). The *Enroll* includes both IP and MAC addresses for the multicast group. Intermediate nodes between R and GW2 create a multicast forwarding cache entry (MFCE, as specified in the SMM-A) to forward multicast data packets for the group by using a MAC address. After receiving the *BSM*, GW2 has already obtained the SP address (i.e., the GW1's IP address) for the multicast group from the received *BSM*, and sends a *Join* message to GW1. The *Join* message has a multicast IP address for the group, and intermediate nodes forward the message and put an entry in the multicast routing table (which is specified as the multicast tree construction mechanism in PIM-SM). Having received the *Join* message, GW1, if it has not received multicast data packets for the group, sends a *Page* message (specified CL2) to all BSs in the paging area where S is. The *Page* contains both IP and MAC addresses for the unicast and multicast. Intermediate nodes forward the *Page* message to downstream interfaces according to the paging cache table with MAC addresses.

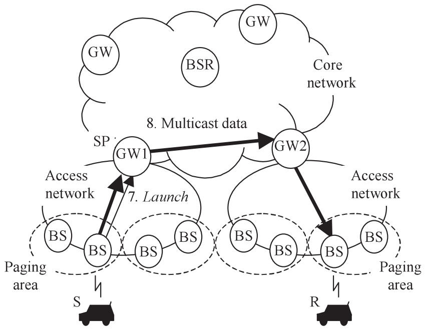

Fig. 8 Network-level procedure 3: Upon receiving a *Page* message, the source host replies to GW1 with a *Launch* message and starts sending multicast data. The multicast data are forwarded to the receiver via GW1 and GW2.

Upon receiving the *Page* message, S replies to GW1 with a *Launch* message (specified in the SMM-A), as shown in **Fig. 8**. The *Launch* contains a MAC address for the multicast group. Intermediate nodes between S and GW1 create an MFCE (specified in the SMM-A) to forward multicast data packets for the group with the MAC address. After that, S starts sending multicast data packets, and these packets are forwarded to GW1, GW2, and R according to each multicast forwarding cache entry.

For more details of SMM procedures, refer to Section 3.2 of Ref. 2).

4. Self-Organizing Tree Mechanism in the Access Network

One of the biggest problems regarding SMM is the topological constraint in the access network. In SMM, as described in Ref. 2), switch nodes and radio base stations in an access network organize a balanced tree topology whose root is a gateway to the core network. Using a υ-ary tree of depth μ; the network needs υ^{μ} base stations and $\sum_{k=1}^{\mu} \upsilon^{k-1}$ switch nodes. This requirement is attributed to the use of the Cellular IP protocol.

With such a physical tree topology, one link failure introduces a service disruption. The access network thus requires some failure recovery features as a distributed network based on the Internet. In addition, the access network must be able to be freely constructed to cover any road topology and be extended according to continuous development of a road traffic infrastructure. In this paper, we propose a protocol to provide the access network with robustness and flexibility. As shown in **Fig. 9**, only radio base stations constitute an access network

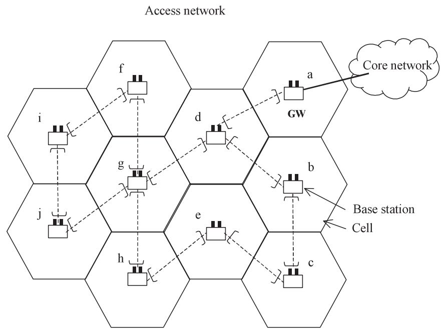

Fig. 9 New access network: the network consists of only radio base stations.

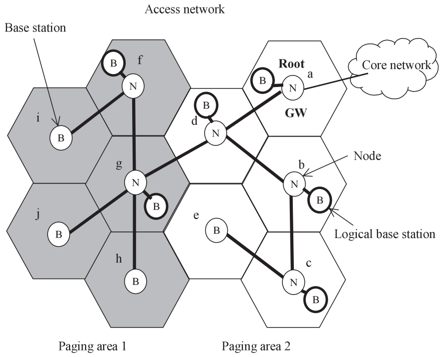

Fig. 10 Logical tree topology in the access network.

with regular hexagonal cells deployment (which minimizes the area of the portion in which cells overlap). They freely connect with each other through wired or wireless links. There is no switching node.

For a completely free topology in an access network, base stations form a logical tree topology, as shown in **Fig. 10**. In such a logical tree, the base stations at the leaves of the tree act as base stations, but those not at leaves act as switch nodes and attach a single virtual base station at their downstream interface. The one that connects to the core network always acts as a switch node as the gateway at the root of the tree. All base stations, including virtual ones, are classified into several paging areas.

Either of two methodologies can be used to allocate paging areas to base stations: static or dynamic assignment. With static assignment, base stations are determined for each paging area before the logical tree is organized, so neighboring base stations can be assigned to the same paging area. However, it can be difficult to construct an optimized tree, that is, a minimum spanning tree. In contrast, with dynamic assignment, paging areas are determined on each leaf after the logical tree has been organized, so the same paging areas are possibly not concentrated so that it introduces frequent procedures of the location registration for a mobile host (in other word, if the same paging areas are deployed sporadically, the host will have to register its location whenever it crosses a paging area boundary).

Table 2 SOT messages.

Message	Parameters	Direction
Hello	Root node's MAC address, accumulated cost, topology change flag	Both
LF-Notification	--	Upstream
P-Notification	Paging area list (ID and the number of leaves)	Upstream
P-Confirmation	Paging area ID	Downstream

Despite the need to frequently register the host's location, we have used dynamic assignment because of the self-healing functionality, namely, a tolerance of unpredictable link status changes.

We will now explain in detail the procedures of the self-organizing tree protocol. The protocol is based on the spanning tree protocol (STP) [23]. **Table 2** shows a list of the messages defined in this protocol.

4.1 Logical Tree Construction

Every base station periodically exchanges *Hello* messages with neighbor base stations. Initially, the state of all interfaces on every base station is the "blocking" state. The root of a tree is always a GW (this is unchangeable). The GW creates root node information '*rootinfo*' and sends it as part of a *Hello* message through all its interfaces. Base stations receiving *rootinfo* set the interface on which *rootinfo* is received as the upstream interface and save the *cost* value included in the *Hello* message. The *rootinfo* is forwarded to other interfaces, which are then set as downstream interfaces. The cost of a link on which the *rootinfo* is sent is added to the *cost* value in the *Hello* message which is sent on the downstream interface. If the *rootinfo* is received on a downstream interface, its *cost* value is compared with the one already saved. If the received *cost* is higher than the saved one, the interface is set to the "blocking" state. (The interface is not used for data packet sending or receiving). Otherwise, the interface is set as an upstream interface and the *rootinfo* is forwarded to other interfaces, which are then set as downstream interfaces. After some time, the state of all interfaces on every base station becomes stable and each interface is an upstream or downstream interface or a blocking

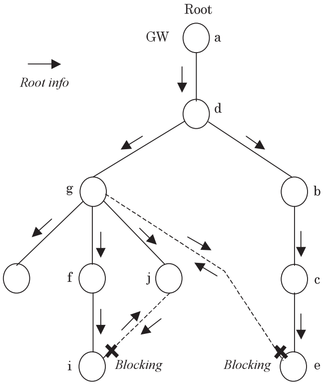

Fig. 11 Procedure 1 of logical tree construction: initially, a simple spanning tree is constructed.

interface. Hence, a logical tree topology whose root is the GW has been organized in the access network. When a link failure is detected by a base station not receiving any *Hello* messages for a predetermined time, topology *change* information is flooded all over the access network through a *Hello* message. Upon receiving the *change* information, each base station will initialize the interface states to "blocking." The above procedure will then be repeated to form a new logical tree.

Figure 11 depicts the initial logical tree formed in the access network from Fig. 9 through the above procedure.

4.2 Paging Area Determination

A base station that has one or more than one downstream interface recognizes that it is not a leaf of the tree and should be a switch node (expressed as N in **Fig. 12**). In this case, it establishes a virtual downstream interface and attaches a virtual base station to the interface (expressed as B with the bold lines in the same figure). This means that the system simultaneously behaves as both a switch node and a base station. Another base station that has no downstream interface recognizes that it is a leaf of the tree and should be a base station (expressed as B in the same figure).

The leaves (including virtual base stations) each send an *LF-Notification* message on their upstream interface. The switch node receiving the *LF-Notification* meters the number of downstream leaves. If the number does not reach a predetermined number, the switch node forwards the *LF-Notification* on its upstream interface. Otherwise, the switch node does not forward the *LF-Notification* and has the paging cache table to manage the host's locations. If the number of leaves on each downstream interface is appropriately large, the switch node assigns different paging areas to each downstream interface. If there is an interface that has a small number of leaves, that interface will be assigned to the same paging area as an interface having a large number of leaves. The switch node having the paging cache table sends a *P-Confirmation* message on its downstream interface, which includes an assigned paging area ID. It also sends a *P-Notification* message on its upstream interface, which includes the number of leaves and assigned paging area ID. A switch node receiving the *P-Confirmation* forwards it through its downstream interface, and the leaf receiving it recognizes the paging area it belongs to and broadcasts a paging area ID to mobile hosts via an air interface. The switch node receiving the *P-Notification* has a paging cache table and merges the paging area information received on the other downstream inter-

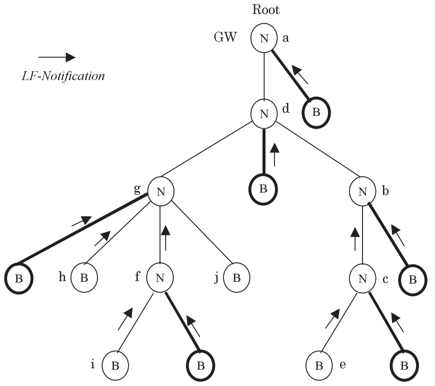

Fig. 12 Procedure 2 of logical tree construction: virtual base stations are attached to non-leaf nodes.

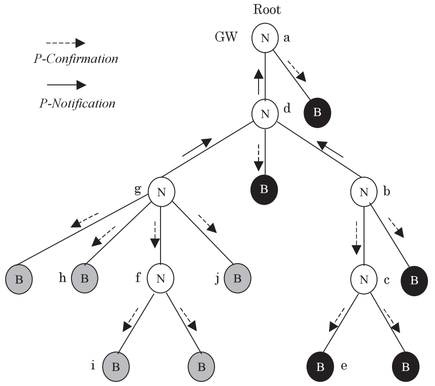

Fig. 13 Procedure 3 of logical tree construction: paging areas are assigned to all base stations.

face and sends it on the upstream interface. If there is an interface having a small number of leaves, it is assigned to the same paging area as one having a large number of leaves. In **Fig. 13**, the base stations shown in gray and black are assigned to different paging areas.

4.3 Considerations Regarding Protocol Performance

SOT protocol inherently has the same constraints of operational scale as STP. The STP specification (Section 8.10 of Ref. 23)) recommends that the maximum network diameter (the maximum number of nodes between any two points on an edge) should be 7, premising that the maximum delay of message transmission and process are respectively 4.0 sec. In the light of the recent increase in computational capacity, we can assume a much smaller process delay and allow the network diameter to be larger. The timer values for state transition can be derived from the maximum network diameter and message transmission and process delays, and a convergence speed can be determined for the construction of logical tree topology. The operational scale (network diameter) and the convergence speed (recovery time from network failure) are in a trade-off relation, and would be determined by the policy of the network service provider.

5. Software Implementation Model

To show how the proposed scheme can be implemented, we next describe the design of a software implementation model. **Figure 14** shows the software architecture of the system for a radio base station, switch node, and gateway to the core network.

When the system functions as a radio base station or switch node (but not a gateway) in an access network, the system implements several physical layer control (PHY) and media access control (MAC) modules for each corresponding network interface (to a mobile host or to some other radio base station), and an SOT that manages all network interfaces to organize a logical tree topology and two CL2 and SMM-A pairs that provide both unicast and multicast forwarding for a radio base station and switch node, respectively. The coordination module shown in the figure provides inter-working functionality between the SOT and CL2/SMM-A. If the SOT determines that the system acts as a switch node, two CL2 and SMM-A pairs simultaneously become active for both a virtual base station and a switch node. Otherwise, a single CL2 and SMM-A pair becomes active for the base station. When protocol packets such as *LF-Notification* are forwarded from a virtual base station to a switch node, the packets are directly forwarded through the SOT rather than sent and received through the MAC and PHY modules.

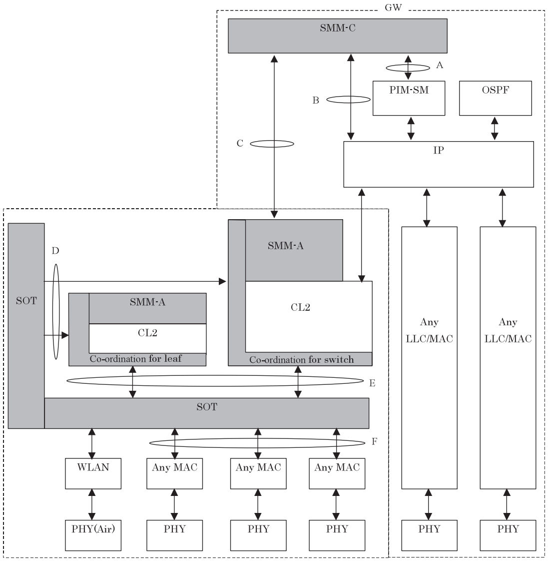

Fig. 14 Software architecture.

When the system functions as a gateway, SMM-C, standard MAC and IP stack and upper routing protocol modules are added. SMM-C provides inter-working functionality to forward multicast data packets between the core and access network and to page multicast sources in the access network, and it also handles the handover mechanism inter-access network. Multicast data sent to or from the core network are forwarded through interface A, but encapsulated packets sent during handover are forwarded through interface B.

In Fig. 14, the modules of the protocol proposed in this paper and those described in Ref. 2) are shaded and the interfaces between such modules are defined as A, B, . . ., F. **Table 3** shows the signals on each of these interfaces, which can be events invoking state transitions in the state machines of each new protocol described below.

Figure 15 shows the state machine or procedure for constructing a logical tree topology in an SOT module. This state machine is implemented for each network interface. Initially, every interface state is in the “disable” state.

Table 3 Definition of inter-module signals (a).

Table 3 Definition of inter-module signals (b).

IF	*Signal* (*event*)	Direction	Description
A	*Data.req*	SMM-C->PIM	This requires forwarding of multicast data to the core network.
	Report.req	SMM-C->PIM	This requires PIM to join a multicast group in the core network (this signifies reception of *Enroll*, which is equivalent to IGMP Report).
	Leave.req	SMM-C->PIM	This requires PIM to leave a multicast group in the core network (this indicates that there is no member in the access network).
	Data.ind	SMM-C<-PIM	This indicates the reception of multicast data to be forwarded to the access network.
	Comp.ind	SMM-C<-PIM	This indicates a new SPT has been constructed in the core network (and implies that SMM-C tunneling should be stopped).
	Delete.ind	SMM-C<-PIM	This indicates there has been no join to an old source point in the core network.
B	*Data.ind*	SMM-C<-IP	This indicates the reception of encapsulated multicast data through SMM-C's tunnel.
	Data.req	SMM-C->IP	This requires transmission of encapsulated multicast data through SMM-C's tunnel.
C	*Data.ind*	SMM-A->SMM-C	This indicates the reception of multicast data to be forwarded to the core network..
	Data.req	SMM-A<-SMM-C	This requires forwarding of multicast data to the access network.
	Page.req	SMM-A<-SMM-C	This requires paging of a multicast source to urge it to send multicast data.
	Enroll.ind	SMM-A->SMM-C	This indicates reception of a request to receive multicast data. If there is no multicast source in the same access network, it triggers the sending of PIM Join to the core network.
	Launch.ind	SMM-A->SMM-C	This indicates that a multicast source should start sending multicast data or indicates the advent of a multicast source from another access network. In the latter case, it triggers inter-access network handover.
	Abort.req	SMM-A<-SMM-C	This requires that forwarding of multicast data to the core network should stop (e.g., there has been no join to an old source point in the core network).
D	*Mode.req*	SOT ->SMM-A	This designates whether a system should behave as a leaf node. If a system becomes a leaf it invokes a leaf only SMM-A entity. Otherwise, it invokes both SMM-A entities (to behave as both a switch and a logical base station)
	Change.req	SOT ->SMM-A	This requires that the state be changed to the initial state because of a topology change.
E	*Data.ind*	SMM-A<-SOT	This indicates the reception of multicast data through a network interface.
	Data.req	SMM-A->SOT	This requires transmission of multicast data through a network interface.
	Msg.ind	SMM-A<-SOT	This indicates the reception of an SMM-A message (*Launch*, *Enroll*, *Move*, *LF-Notification*, *P-Notification*, etc.).
	Msg.req	SMM-A->SOT	This requires transmission of an SMM-A message (*Launch*, *Enroll*, *Move*, *LF-Notification*, *P-Notification*, etc.).
F	*Data.ind*	SOT<-MAC	This indicates the reception of multicast data through a network interface
	Data.req	SOT->MAC	This requires transmission of multicast data through a network interface.
	Hello.ind	SOT<-MAC	This indicates the reception of a *Hello* message including *Rootinfo* or *Change*.
	Hello.req	SOT->MAC	This requires transmission of a *Hello* message including *Rootinfo* or *Change*.

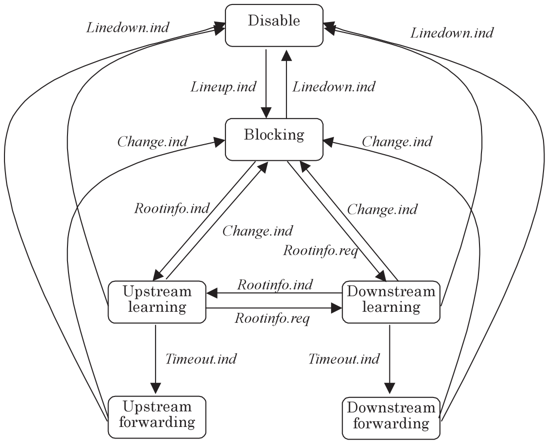

Fig. 15 State machine on an SOT (construction of a spanning tree).

Upon detecting the line-up status of the interface, the state changes to "blocking." Receiving or sending *rootinfo* through the interface causes the state to change to "upstream learning" or "downstream learning," respectively. During these states, the interface is not yet allowed to send and receive data packets (the states will flap for a time until the logical tree has been constructed). After a predetermined time has elapsed (a system timer expires), these states change to "upstream forwarding" or "downstream forwarding," respectively. In these states the system can send and receive data packets on the interface. If the system detects a line-down status at the interface, the state of the interface changes to "disable" and the state of the other interfaces changes to "blocking," and then the system floods topology *change* information on the other interfaces through a *Hello* message. When the system receives topology *change* information on one of the interfaces, the state of all interfaces changes to "blocking."

Figure 16 shows a state machine on the SOT which manages whether the system will be a switch node or a leaf. Only one such

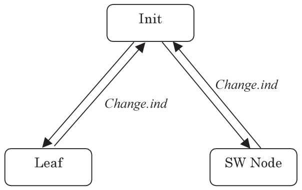

Fig. 16 State machine on an SOT (management of system behavior).

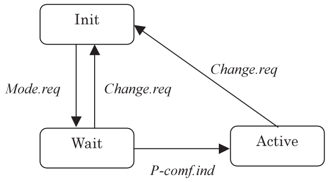

Fig. 17 State machine on SMM-A/CL2 coordinate module.

state machine is implemented in the system. Initially, the state is "init." When every network interface settles in one of the above states ("disable," "blocking," "upstream forwarding," or "downstream forwarding"), the SOT determines whether the system will become a switch node or a base station, and the state then changes to "SW node" or "leaf." At this time, the SOT issues a *Mode.req* signal to the coordinate module of SMM-A/CL2. Whenever a topology change occurs, the state goes back to "init."

Figure 17 shows the state machine on the coordinate module of SMM-A/CL2, which performs the paging area allocation function. Two state machines are implemented: one each for the switch node and base station. The state is initially "init." In the case of a leaf, upon receiving *Mode.req* from the SOT, the coordinate module sends *LF-Notification* through the upstream interface and sets the state to "wait." After *P-Confirmation* is received through an upstream interface, the state changes to "active." In the case of a switch node, when *Mode.req* is received from the SOT the state changes to "wait." When the decision is made to have a paging cache table after receiving *LF-Notification* through a downstream interface or when *P-Confirmation* is received through an upstream interface, the state changes to "active." When the state is not "active," the SMM-A and CL2 modules ignore any control packets for paging and routing, and discard any received data packets. Whenever a topology change takes place, the state goes back to "init."

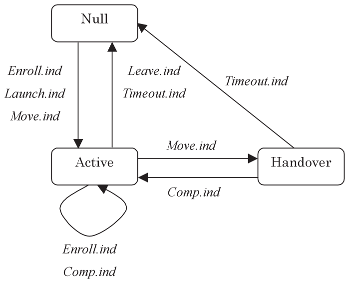

Fig. 18 State machine on SMM-A.

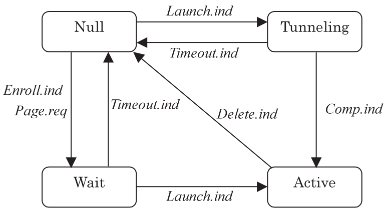

Fig. 19 State machine on SMM-C.

Figure 18 shows the state machine on SMM-A that provides multicast packet forwarding and intra-access network handover (refer to Section 3.2.1 in Ref. 2)). The state machine is managed through multicast forwarding cache entry (MFCE) for each multicast group. Initially, the state is "null" (there is no entry). Upon receiving an *Enroll*, *Launch*, or *Move* message, SMM-A creates a new entry for the multicast group and the state changes to "active," during which time multicast data packets can be forwarded. If SMM-A receives a *Move* message in the "active" state, the state changes to "handover" and another entry with a transition flag set is created for the multicast group. Every entry will be removed when a *Leave* message is received or the aging timer expires.

Figure 19 shows the state machine on SMM-C, which provides forwarding multicast packets to the core network and a tunneling mechanism for inter-access network handover (refer to Section 3.2.2 in Ref. 2)). The state machine is managed for each multicast group. Initially, the state is "null." Upon receiving *Enroll* or PIM *Join* from the core network in a "null" state, SMM-C issues a *page.req* signal to SMM-A to page a multicast source, and the state changes to "wait." After receiving *Launch* from the multicast source during the "wait" state, the state changes to "active." This state signifies that the rendezvous point in the access network (and also the source point in the core network) is receiving multicast data from a multicast source. Meanwhile, upon receiving *Launch* from another access network in the 'null' state, the state changes to "tunneling" and SMM-C establishes a tunnel to a previous source point in the core network (i.e., it sends encapsulated multicast packets to the IP instead of PIM-SM). If a *Comp.ind* signal is received from PIM-SM during "tunneling," the state changes to "active."

6. Conclusion

We have proposed additional functionalities and improvements for the Future Vision Distribution Service and the Source Mobility Support Multicasting technique. This paper further supports the feasibility of the Future Vision Distribution Service on an ITS, in line with our previous proposals. The main contributions in this paper are as follows.

First, this paper has described an application-level protocol, the Mobile Vehicle Management (MVM) protocol, for use between a client in a vehicle and servers in the network to update the current position and traveling route of each vehicle and provide a preceding vehicle's multicast address. We have explained in detail how the system finds a preceding vehicle moving at an appropriate position along the same traveling route and provides this vehicle's multicast address. The separation of these functionalities and network-level functionalities allows us to avoid increasing the complexity of the network-level protocol.

Second, this paper has proposed a network-level protocol, the self-organizing tree (SOT) protocol, for use among nodes in the access network, to enable the nodes to construct a logical tree topology in a self-forming and self-healing manner by using a spanning tree algorithm. The SOT provides the network with robustness from link and node failures and flexibility for network design to cover any road topology. As a result, it allows radio base-stations to organize themselves without any switch nodes. This protocol could eliminate the topological constraint in the access network, which is one of the biggest problems in the implementation of the proposed service.

Third, this paper has described in detail a software implementation model. We have designed a software architecture, the composition elements for each protocol and their relations, and internal state-machines for each element. Hence, we have shown how FVDS and SMM can be implemented.

In future studies, we need to define more detailed protocol specifications such as packet format, timers, and operational parameters. We plan to consider QoS guarantee issues on SMM schemes, since FVDS features real-time video distribution that strictly requires a secure bandwidth. We also plan to study security control and an authentication mechanism for users joining this service.

References

1) Sato, K., Katsumoto, M. and Miki, T.: A New Multicast Technique Supporting Source Mobility for Future Vision Delivery Service, *Proc. 18th Int'l Conf. on Advanced Information Networking and Applications* (Mar. 2004).
2) Sato, K., Katsumoto, M. and Miki, T.: Source Mobility Multicast (SMM), *IPSJ Journal*, Vol.45, No.2 (2004).
3) Internet ITS Project: Building Internet ITS Infrastructure. http://www.internetits.org/en/infra2.html
4) http://www.meshnetworks.com/pages/solutions/its.htm#
5) http://www.oki.com/jp/Home/JIS/New/OKI-News/2004/02/z03096.html
6) Munaka, T., Yamamoto, T., et al.: Advanced Join Mechanism for Multicast Group Management in ITS Networks and Its Evaluation, *IPSJ Journal*, Vol.42 No.7 (2001).
7) Munaka, T., et al.: A Reliable Multicast Mechanism for Location Dependent Data in DSRC Based Networks, *IEICE Trans.*, Vol.85-D, No.11 (2002).
8) Moy, J.: OSPF Version 2, RFC2328 (Apr. 1998).
9) Estrin, D., Farinacci, D., et al.: Protocol Independent Multicast-Sparse Mode, RFC2362 (1998).
10) Fenner, B., Handley, M., et al.: Protocol Independent Multicast-Sparse Mode (Revised) IETF Internet Draft (Feb. 2004).
11) Perkins, C., et al.: IP Mobility Support, RFC2002 (Oct. 1996).
12) Perkins, C. and Johnson, D.B.: Route optimization in Mobile IP, IETF Internet Draft (Feb. 1999).
13) Koodli, R., et al.: Fast Handovers for Mobile IPv6, IETF Internet Draft (Sep. 2002).
14) Valko, A.G.: Cellular IP: A New Approach to Internet Host Mobility, *ACM Computer Communication Review* (Jan. 1999).
15) Zach, D., Gatzounas, D., Campbell, A., et al.: Cellular IPv6, IETF Internet Draft (July 2001).
16) Gatzounas, D., Malataras, A., et al.: Route

Optimization for Cellular IP Networks, *Proc. IPNC 2001*, France (May 2001).
17) Harrison, T.G., Williamson, C.L., et al.: Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts, *Proc. ACM/IEEE MOBICOM '97*, Budapest, Hungary (Sep. 1997).
18) Wang, Y. and Chen, W.: Supporting IP Multicast for Mobile Hosts, *ACM/Kluwer Mobile Networks and Applications, Special Issue on Wireless Internet and Intranet Access*, Vol.6, No.1 (2001).
19) Lin, C.R. and Wang, K.M.: Mobile Multicast Support in IP Networks, *Proc. IEEE INFOCOM '00*, Tel Aviv, Israel (Mar. 2000).
20) Sung, S., Shin, Y., et al.: Mobile Multicast Method Using the Habitude of Movement, *16th Intl' Conf. on Information Network*, Chez, Korea (Jan. 2002).
21) Rolland, V., Luis, C. and Serge, F.: Mobility and Multicast: Protocol Design and Analysis, *17th Intl' Symposium on Computer and Information Science*, Orlando, FL (Oct. 2002).
22) Laubach, M. and Halpern, J.: Classical IP and ARP over ATM, RFC2225 (Apr. 1998).
23) Media Access Control (MAC) Bridges, IEEE Std 802.1d (1998).

(Received May 11, 2004)
(Accepted September 3, 2004)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.91–103.)

Katsuhiko Sato received the B.E. degree from Aoyama Gakuin University in 1992, and the M.E. degree from the University of Electro-Communication in 2002. He has been working for Japan Radio Co., Ltd., where he has developed Frame Relay Router, ATM Switch, 3G Wireless Base station and Multi-layer Switch, and he is currently developing Wireless Mesh Router. He was a visiting researcher at Communication Research Laboratory in 2002–2003. He is currently working toward the Ph.D. in the University of Electro-Communications. His research interests include mobile and multicast routing and VOD system. He is a member of IPSJ, IEEE Communication Society.

Michiaki Katsumoto received the B.S., M.S. and Ph.D. degrees from Toyo University in 1991, 1993 and 1996 respectively. He is working now National Institute of Information and Communications Technology. His research interests include next generation Internet applications and high-quality multimedia contents. He is a member of IPSJ, IEICE, IEEE Computer Society and ACM.

Tetsuya Miki received a B.E. degree from the University of Electro-Communications in 1965, and M.E. and Ph.D. degrees from Tohoku University in 1967 and 1970, respectively. He is currently a Professor at the Department of Information and Communication Engineering, the University of Electro-Communications. His research involves optical and wireless access systems, photonic networks, multimedia information communications, and network architectures. He joined the Electrical Communication Laboratories of NTT in 1970, where he was engaged in the research and development of high-speed digital transmission systems using coaxial cable, optical transmission systems, optical access networks, ATM and multimedia transport networks, network operation systems and network architecture. He was the Executive Manager of the NTT Optical Network Systems Laboratories in 1992–1995. He received the IEICE Achievement Award in 1978 for his contribution to the early development of optical transmission systems. He is a Fellow of IEEE, and was Vice-President of IEEE Communications Society during 1998 and 1999. He is also a Fellow of IEICE, and is currently Vice-President of IEICE.