An ATM Network Management System for Point-to-Multipoint Reservation Service

Daniel Won-Kyu Hong, Choong Seon Hong, Dong Sik Yun,
Jae Hyoung Yoo, and Woo-Sung Kim

This paper describes an integrated network management framework for providing point-to-multipoint reservation service (PMRS) in an ATM network. There are two major issues confronting the network service provider in relation to this service: one is to rapidly confirm the acceptability of the subscriber's reservation at subscription time, and the other is to punctually activate the reserved point-to-multipoint service. To meet these requirements, we developed a service provision model (SPM) and a network resource model of a bandwidth allocation timetable (BATT). We propose a point-to-multipoint routing algorithm composed of ordering and backtracking procedures, which can find the best branch point under the complex network topology and can add more destinations to the existing point-to-multipoint route. We demonstrate the feasibility of the SPM, the BATT, and the point-to-multipoint routing algorithm by implementing our schemes and analyzing their performance under the operational ATM network of KT(Korea Telecom).

I. INTRODUCTION

In industries such as distance learning systems, newspaper publication and health care, there is an ever-increasing demand for high-quality point-to-multipoint applications which can offer bulk data transfer from one point to multiple branches in an ATM network [1]-[10]. Consider the distance learning system. It is not a long-lasting service; it requires short but recurrent service. In the case of a newspaper publishing company, the head office edits the newspaper with all its news reports, articles, photographs, and advertisements and distributes the newspaper electronically to its subsidiaries. In addition, the promise of telemedicine to improve healthcare delivery is increasingly becoming a reality. Today, hospitals use ATM networks to provide imaging quality and transmission speeds that allow surgeons in an operating room to consult visually with specialists not present during an operation. Clarity and low latency is the key for MRI and other image interpretation functions. A provision of point-to-multipoint reservation service is necessary for these applications because customers who want to use such applications usually prepare for their use in advance.

This paper describes an integrated network management framework for providing point-to-multipoint reservation service (PMRS) in an ATM network. From the perspective of a network provider, there are two major issues in providing the PMRS: one is to rapidly confirm the reservation service during subscription time, and the other is to punctually activate the reserved service. Our point-to-multipoint reservation model aims to confirm the availability of the reservation service to subscribers while they are on the line. Therefore, a mechanism to minimize notification time is crucial.

Manuscript received Oct. 31, 2001; revised May 30, 2002.

Daniel Won-Kyu Hong (phone: +82 42 870 8254, e-mail: wkhong@kt.co.kr), Dong Sik Yun (e-mail: dsyun@kt.co.kr), Jae Hyoung Yoo (e-mail: styoo@kt.co.kr), and Woo-Sung Kim (e-mail: wskim@kt.co.kr) are with KT, Daejeon, Korea.

Choong Seon Hong (e-mail: cshong@khu.ac.kr) is with the School of Electronics and Information, Kyung Hee University, Yongin, Kyungki-Do, Korea.

It is necessary for a network provider to adopt a new network resource management model in order to rapidly confirm a subscriber’s reservation service request during the time of subscription. This paper proposes a network resource management model based on a bandwidth allocation timetable (BATT) scheme that can confirm network resource availability for a requested time period while maximizing network resource utilization.

There are two research issues concerned with providing point-to-multipoint service in an ATM network: one is a routing algorithm for determining the best branch point in a complex network topology, and the other is an algorithm for adding more destinations to the existing point-to-multipoint route while still providing reasonable computational performance. To meet these requirements, we propose a point-to-multipoint routing algorithm composed of ordering and backtracking procedures. Lastly, this paper illustrates the feasibility of our integrated network management system composed of the service provision model (SPM), the BATT, and the point-to-multipoint routing algorithm by implementing and analyzing their performance under the ATM network of KT.

The rest of this paper is organized as follows: in the next section, we present the SPM for point-to-multipoint service in an ATM network and describe the network resource management model based on the BATT. Section III illustrates the point-to-multi-point routing algorithm composed of ordering and backtracking procedures including a method under which destinations to the existing point-to-multipoint route are added. Section IV describes the implementation model and the performance analysis results. Section V discusses the merits and issues of the integrated network management framework for providing point-to-multipoint service in an ATM network and then gives some concluding remarks.

II. SERVICE PROVISION AND NETWORK RESOURCE MODELS

This section describes our service provision model and point-to-multipoint routing algorithm for providing the point-to-multipoint reservation service in ATM networks.

1. A Service Provision Model (SPM)

To provide point-to-multipoint reservation service in an ATM network, we need an SPM from the perspective of the network service provider (Fig. 1). There are two kinds of communication modes, permanent and reserved. In the permanent communication, the subscriber can negotiate with the network service provider (NSP) any value for the peak cell rate (PCR) that is available at the user-network reference point (UNRP) and is agreed upon by the NSP. To establish reserved communication, the subscriber can negotiate with the NSP any value of the maximum bandwidth that is available at the user to network interface (UNI) and is agreed upon by the NSP. The NSP can then negotiate with the network management system (NMS) any available value of the PCR within the subscribed maximum bandwidth. The NSP enforces user cells according to the

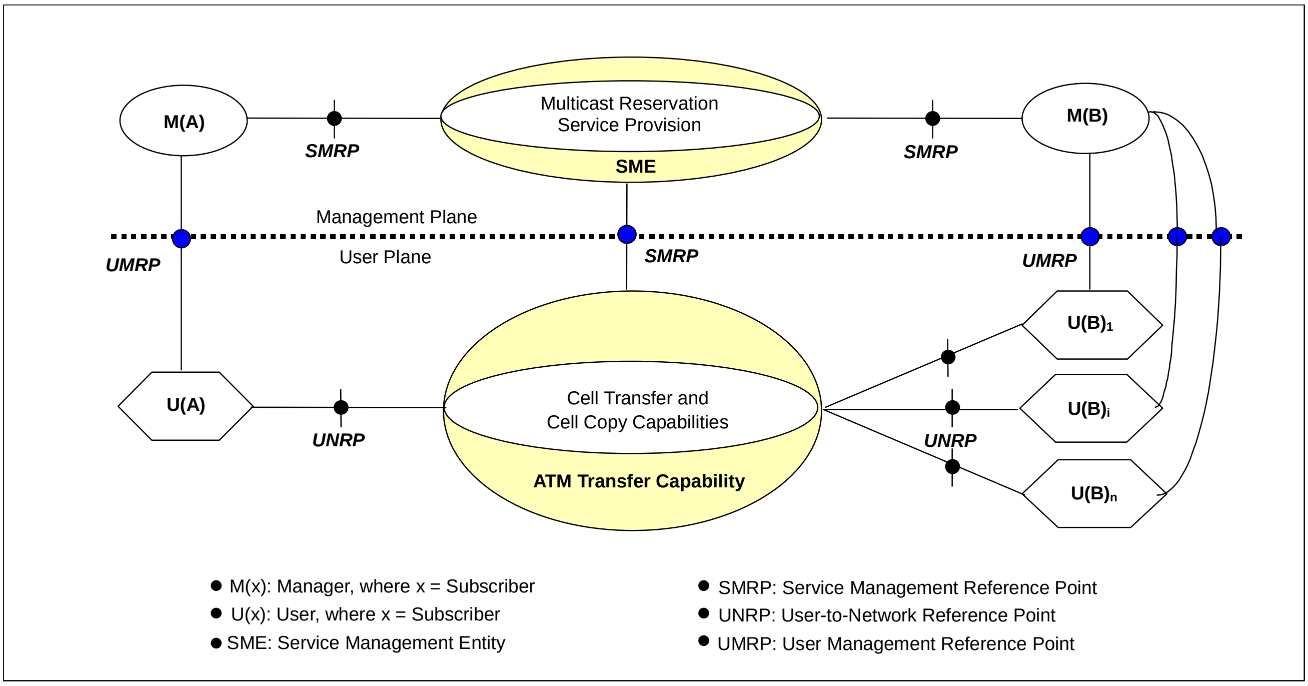


Fig. 1. A service provision model for the point-to-multipoint reservation service.

PCR reference algorithm specified in Recommendation I.371 [1]. According to the PCR reference algorithm [1], cell conformance is defined with reference to the pair (T, τ) where T is the peak emission interval (PEI) and τ is the cell delay variation (CDV) tolerance [3]. The NSP provides the user with PEI values and CDV tolerance as part of the traffic contract.

Periodic or non-periodic permanent virtual connections (PVCs) are possible in the reserved establishment of communication. The subscriber uses the PMRS in the reserved establishment of communication providing general service parameters, such as a manager identifier, users E.164 [11] number, subscription beginning time (t_0), subscription end time (t_1), and maximum number Nb_{max} of PVCs per UNRP and maximum bandwidth (Nb_{max}) needed at each UNRP. E.164 typically consists of decimal digits segmented into groups for identifying specific elements for identification, routing, and charging capabilities, e.g., within E.164 to identify countries, national destinations, and subscribers [11]. All reserved PVCs are established after the registration procedure (Fig. 2).

This paper focuses only on non-periodic reservation service. During this phase the parameters of the PVCs among the involved users are negotiated. The PMRS is provided after arrangement with the NSP. The PMRS permits the reserved or permanent communication between two or more UNIs. Subscriptions of reserved and permanent communication can coexist on the same UNI. The subscription states the maximum number Nb_{max} of PVCs per UNI. The number of PVCs available at the UNI is less than or equal to 256. Hence, Nb_{max} is evaluated taking into account these constraints.

The notification time for confirming the reservation service request $T_N = t_1 - t_0$ will be larger than or equal to a minimum notification time of $T_{N_{min}}$. The service availability $S_A = t_5 - t_2$ corresponds to the duration of reservation service availability. S_A varies between the minimum duration of subscription $S_{A_{min}}$ and the maximum duration of subscription $S_{A_{max}}$. If t$_5$ is not indicated, we assume S_A is equal to $S_{A_{max}}$. The values $T_{N_{min}}$, $S_{A_{min}}$, and $S_{A_{max}}$ are fixed by the NSP. We aim to confirm the availability of the reservation service to subscribers while they are online. Therefore, a mechanism to minimize notification time is crucial. We will propose it in accordance with the implementation and performance analysis results found in section IV.

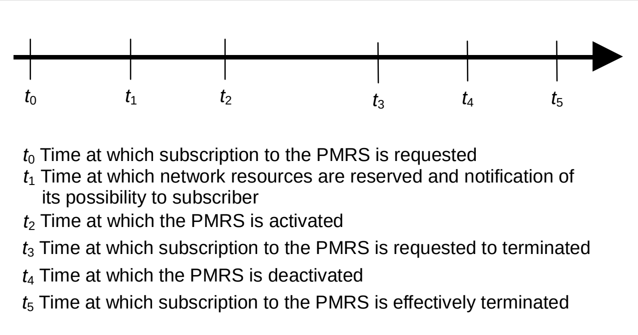


Fig. 2. PMRS provision and withdrawal procedures.

2. A Network Resource Management Model

To support the two previously described concepts—rapidly validating the service availability at the time of notification of T_N and punctual guarantee of service availability once the NSP has agreed to provide the reservation service with the customers—this paper proposes a network resource management scheme from the perspective of an ATM network management system (NMS) (Fig. 3). Network topology

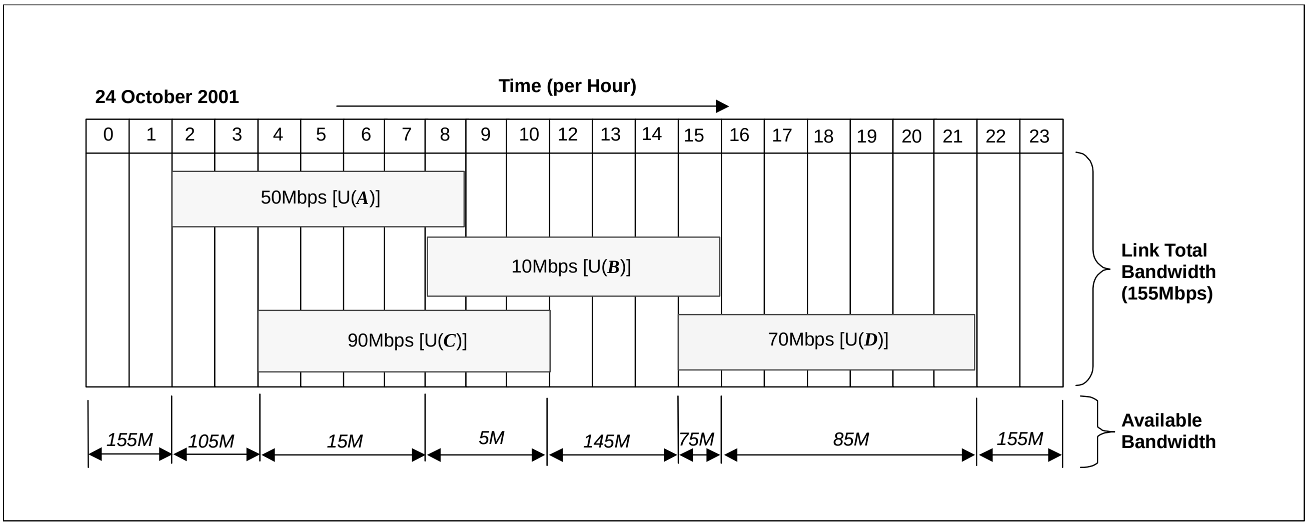


Fig. 3. Bandwidth allocation timetable (BATT).

consists of a node and a link. The NMS must find the best point-to-multipoint route traversing the complex network topology, including the connection admission control (CAC) [12], in order to determine the service availability at the time of notification T_N. For confirming the network resource reservation prior to service activation, all links in the network topology are included in the BATT (Fig. 3).

We limit the minimum period of reservation service to an hour for the purpose of simplifying the procedure. The minimum BATT is one day. When a user subscribes to the reservation service, the NMS determines whether the subscription can be accommodated by consulting the BATT. When no BATT corresponds to the requested reservation period, for example from 8:00 to 15:00 on 24 October 2001, the NMS adds a new BATT corresponding to 24 October 2001. If a user subscribes to the reservation service with 10 Mbps from 08:00 to 20:00 on 24 October 2001 under the situation shown in Fig. 3, the reservation request will be rejected because of lack of available bandwidth. Thus, the NMS can determine whether it can accommodate the requested reservation service within the notification time T_N.

III. A POINT-TO-MULTIPOINT ROUTING ALGORITHM

One of the important research issues in point-to-multipoint routing is determining where the best branch point is (Fig. 4). This figure shows one source node (A) and three destination nodes (E, F, and J). If we assume that all link costs are equal, there is no doubt that node D is the first branch point. However, there are 36 candidate routes to connect from node D to the destination nodes E, F, and J. If the destination node is only J, node G will be the next branch point. It is not easy to determine which node is the best branch point to reach multiple destination nodes and still maximize network resource utilization and minimized transit delay. This is because there are two destination nodes, E and F, other than J.

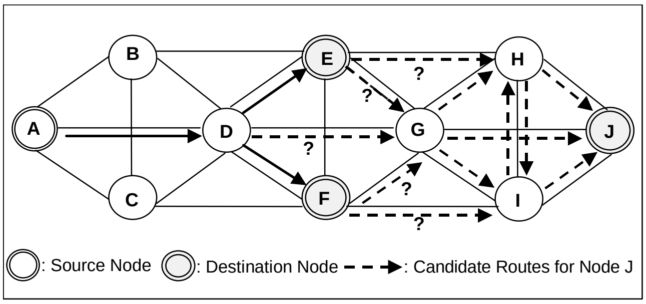


Fig. 4. Best branch point determination problem.

There are two point-to-multipoint routing models: one is for finding the point-to-multipoint route immediately [13]-[15] and the other is for adding one or more branches to existing point-to-multipoint connections [13]-[15]. To remove the difficulty in determining the best branch point and support these two point-to-multipoint models, we propose a point-to-multipoint routing algorithm composed of ordering and backtracking procedures that can identify the best branch points under a complex network topology and still offer reasonable performance.

1. Ordering Procedure

The ordering procedure is for determining the orders of links and nodes taking the routing metrics of the available bandwidth delay and hop count into account. Figure 5 shows the ordering procedure for finding the best point-to-multipoint route from source node A to destination nodes of B, D, and H with a requested bandwidth of 20 Mbps.

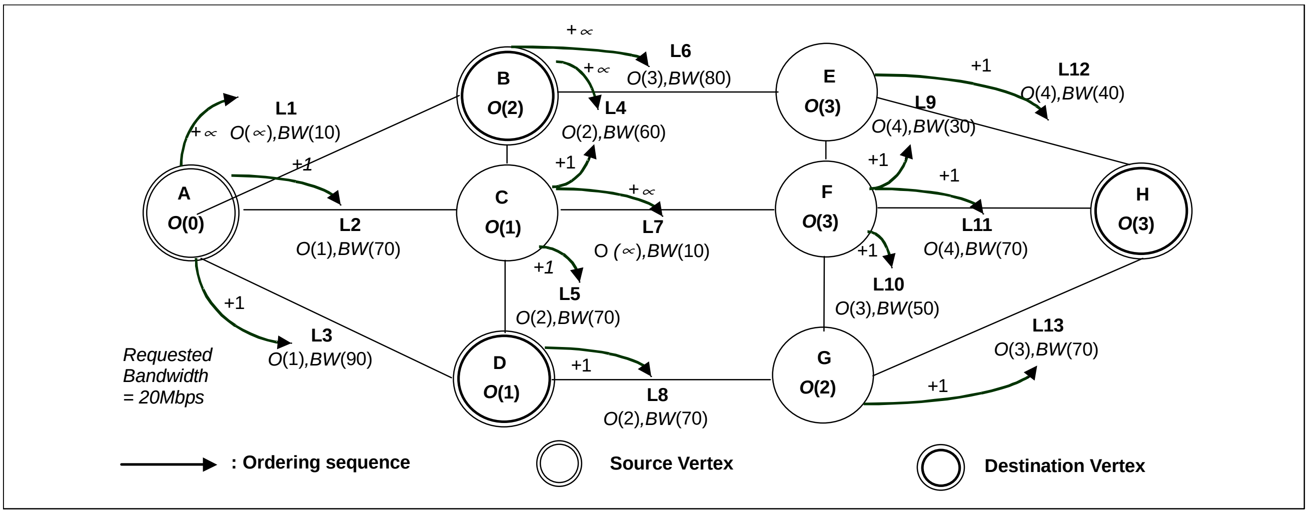


Fig. 5. Ordering procedure.

The network topology is represented by Graph $G(V,E)$ where Vertex (V) denotes a set of nodes and Edge (E) denotes a set of links among nodes. Vertex contains three kinds of information: order (O), visit flag ($VISIT$), and availability (AVA); edge maintains three kinds of information: order (O), available bandwidth (B_{AVA}), and availability (AVA). Availability represents the possibility of the cell transfer capability of each vertex or edge, which depends mostly on fault status or performance degradation of nodes or links.

Figure 5 shows a sample network topology and the ordering procedure. The arrows represent the topology traversing sequence for order assignment. The infinite order (∝) represents the edges of the unusable vertexes because of a fault or lack of available bandwidth. *BW* represents the available bandwidth of the edge, and "+1" represents the increment of order by one. There is one source vertex *A* and three destination vertexes, *B*, *D,* and *H.* We traverse the network topology to determine the order of vertexes and edges. We define two terms: the visiting vertex (V_{visit}) and the neighbor vertex (V_{neigh}). The V_{visit} is *an active vertex* that allocates the orders of all neighbor vertexes (V_{neigh}) connected to it and sets its visit flag to *YES*. On the other hand, V_{neigh} is *a passive vertex* in that its order is determined by the order of the edge connected to it and its visit flag is not changed to *YES* until it can be V_{visit}. For example, in Fig. 5 when we start to traverse the network topology from source vertex *A*, it can be V_{visit}, and the vertexes of *B, C,* and *D* connected to the edges (*L1*, *L2* and *L3*) of vertex *A* can be V_{neighs}. Figure 6 presents the ordering algorithm, and Fig. 6(b) includes the ordering sequence when the source vertex is *A* and the destination vertexes are *B*, *D*, and *H*. There are eight steps to complete the ordering under the network topology shown in Fig. 5. In Fig. 6(b), the vertexes at the VISIT column are active vertexes while the vertexes at the same level as the active vertexes are the passive vertexes in each ordering step.

```
$G(V,E)$, where $V$ : Vertex,   $E$ : Edge
$V = \{O, VISIT\}$,   $E = \{O, B_{AVA}\}$            // $O$ : Order, $B_{AVA}$ : Available Bandwidth
$B_{REQ}$ : Requested Bandwidth
$E_{ADJ}$ : Adjacent Vertex of E
for all $E(O) \leftarrow$ ∝;                          // ∝ : Infinite
$V_{VISIT}(O, VISIT) \leftarrow$ (0,YES);             // $V_{VISIT}$ : Source Vertex

Procedure **Ordering**($V_{VISIT}$, $B_{REQ}$) {
        for all $E \in V_{VISIT}$ {
                $E(O) \leftarrow V_{VISIT}(O)+1$;
                $V_{NEIGH} \leftarrow E_{ADJ}$;    // Adjacent Vertex of E
                if($V_{NEIGH}(VISIT)$ == No && $V_{NEIGH}(O) < E(O)$ && $E(B_{AVA}) \geq B_{REQ}$) {
                        $V_{NEIGH}(O, VISIT) \leftarrow$ ($E(O)$, YES);
                        if($V_{NEIGH}(O)$ != ∝)
                                **Ordering**($V_{NEIGH}$, $B_{REQ}$);        // Recursive Call Ordering Procedure
                } else {
                        $E(O) \leftarrow$ ∝;
                        $V_{NEIGH}(O) \leftarrow$ ∝;
                }
        } // end of for
} // end of Ordering
```

(a) Ordering Algorithm

STEP	VISIT	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13
Initial Value		∝	∝	∝	∝	∝	∝	∝	∝	∝	∝	∝	∝	∝
Step1	A(0)	∝,B(∝)	1,C(1)	1,D(1)										
Step2	C(1)		1, A(0)		2,B(2)	2,D(1)		∝,F(∝)						
Step3	B(2)	∝,A(0)			2,C(1)		3,E(3)							
Step4	D(1)			1,A(0)		2,C(1)			2,G(2)					
Step5	E(3)						3,B(2)			4,F(4)		4,H(4)		
Step6	G(2)								2,D(1)		3,F(3)			3,H(3)
Step7	F(3)							∝,C(1)		4,E(3)			4,H(3)	
Step8	H(3)											4,E(3)	4,F(3)	3,G(3)

X, V(Y), where *X* = order of edge, *V* = adjacent vertex, *Y* = order of adjacent vertex

(b) Ordering Result

Fig. 6. Ordering algorithm.

The ordering procedure is as follows:

Initially, we assign infinite (∝) to all orders of vertexes and edges and assign *NO* to the visit flags of all vertexes (Step 1 of Fig. 6(b)). We assign zero to the order of source vertex *A*, which can serve as the first V_{visit}.

From here on, we traverse the network topology with the ordering algorithm of Fig. 6(a) until the visit flag of all vertexes is set to *YES* and the order of all edges and vertexes according to hop count, availability, and available bandwidth is assigned.

The determination of the next V_{visit} is done in the way of the depth first search. It is normal for one V_{neigh} to serve as the next V_{visit} . Here, we define one rule where the V_{neigh} can serve as the next V_{visit} because its order is infinite (∝). For example, in step 2 of Fig. 6(b), the next V_{visit} vertex is not *B* but *C* because the order of *B* is infinite (∝) in step 1.

In the process of traversing the network topology, the order is assigned according to the simple rule that the order of edges connected to the V_{visit} is assigned by the order of the V_{visit} plus one $(E(O) \leftarrow V_{visit}(O)+1)$.

If the order of any edge is larger than the order of V_{visit} plus one, it is changed to the order of V_{visit} plus one. If the order of any edge is less than or equal to the order of V_{visit} plus one, it will not be changed. For example, if V_{visit} is *A*, the links connected to vertex *A* are *L1*, *L2*, and *L3*. The requested bandwidth (B_{REQ}) is 20 Mbps. The order of the edges connected to vertex A is replaced with one (the order of vertex A plus one) because the order of the first V_{visit} *A* is zero. If any link can accommodate B_{REQ} and its order is less than one, its order will be changed to one as with *L2* and *L3*. However, if any link cannot accommodate B_{REQ}, its order is changed to infinite (∝), having nothing to do with the order of V_{visit}. If the order of any link is greater than or equal to the order of the V_{visit} plus one, as with *L1*, it will not be changed. Figure 6 shows the detailed routing algorithm and its ordering result.

2. Backtracking Procedure

Backtracking aims to determine the best point-to-multipoint route based on the order information allocated in the process of ordering. In backtracking, the source and destination vertexes are the reverse of what they were in the ordering procedure. In addition, the CAC on the selected edges is done in the backtracking procedure. Figure 7 shows the backtracking procedure, which is based on the ordering result of Fig. 6. In the backtracking procedure, there are three source vertexes, *B*, *D* and *H*, and one destination vertex, *A*.

When we traverse the network topology from the source to the destination vertexes, we select the edge having the lowest order. For example, source vertex *B* selects *L4* and the best edge among the edges of *L1*, *L4*, and *L6* because the order of *L4* has the lowest order. Thus, the CAC on the selected edges is carried out in backtracking. The intermediate vertex *C* selects *L2* among the edges of *L2* and *L5* because *L2* has the lowest order. Therefore, the best route between vertex *A* and vertex *B* is <*A-L2-C-L4*>. Furthermore, source vertex *H* selects *L13* as a best edge in the same way that vertex *B* does. The intermediate vertex *G* and *D* selects the edges having the lowest order among the possible edges. As a result of backtracking from vertex *H* to vertex *A*, the best route found is <*A-L3-D-L8-G-L13-H*>.

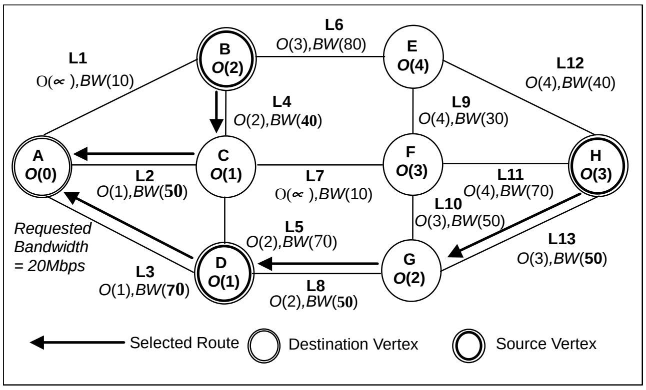


Fig. 7. Backtracking procedure.

If there are two or more edges with the same order, the edge having the largest residual bandwidth is selected. In the process of backtracking, the BATTs of edges change according to the B_{REQ} and the reservation period. For example, when vertex *H* selects the lowest order edge of *L13*, the bandwidth of *L13*, 70 Mbps, is changed to 50 Mbps because the requested bandwidth is 20 Mbps. Furthermore, we stop backtracking when we encounter the edges or vertexes through which we have already backtracked in a previous step. For example, when the source vertex is *D* and the destination vertex is *A*, vertex *D* selects the edge of *L3*. Subsequently, when the source vertex is *H* and the destination vertex is *A*, vertex *H* selects *L13* and *L8* as the best edges. Vertex H does not select the edge of *L3* because we have already backtracked in the previous step between source vertex *A* and destination vertex *D*. Thus, the backtracking procedure can determine the best branch points and subsequently provide the best point-to-multipoint route in combination with the ordering procedure.

3. Adding a Branch Procedure

It is not easy to find the best branch points when adding one or more new destination vertexes to existing point-to-multipoint routes because the existing point-to-multipoint routes are already optimized for the designated source vertex and destination vertexes. Our adding branch scheme finds new best branch points by taking into account the existing point-to-multipoint route and changed network status (Fig. 8). The adding branch algorithm also uses the ordering and backtracking algorithms to find new best branch points. However, the newly added scheme differs from the previously described ordering and backtracking algorithms in that the edge has one additional item, an *existing route flag*, which notes whether the edge is involved in an existing point-to-multipoint route. Thus, the edge has three kinds of information: order (*O*), available bandwidth (B_{AVA}), and existing route flag (*R*).

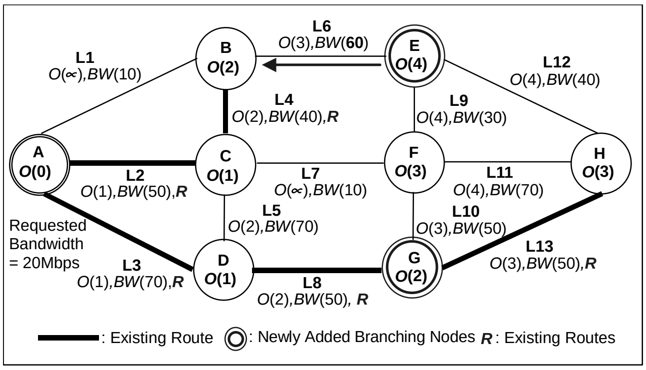


Fig. 8. Adding branch algorithm.

Let us assume that the existing point-to-multipoint route resembles the one in Fig. 7, and the newly adding branch vertexes are *E* and *G* as shown in Fig. 8. The adding branch procedure is as follows.

The first step determines the order of vertexes and edges with the ordering algorithm. At this stage, the differences from the previously described ordering algorithm are as follows: it marks the existing routing flag (*R*) in the edges which are parts of existing routes and it does not consider the B_{AVA} on the edges marked as the existing route flag. For example, because the existing route flags (*R*) of edges *L2, L3, L4, L8* and *L13* are marked, their B_{AVA} are not considered in the ordering procedure. However, the edges other than *L2*, *L3*, *L4*, *L8* and *L13* are not marked so the existing route flag does consider their B_{AVA} in exactly the same way the previously described ordering algorithm.

From the perspective of backtracking for adding branches to existing routes, what is different from the backtracking algorithm described in section III.2 is that it does not adjust the BATTs of the edges, which are parts of the existing point-to-multipoint route. For example, we do not adjust the BATTs shown in Fig. 8 because the existing route flags (*R*) of *L2, L3, L4, L8,* and *L13* are marked. However, the bandwidth of the newly added edge of *L6* in the additional branch must be adjusted. If a new branch *G* is added to the existing point-to-multipoint routes as shown in Fig. 8, the BATT adjustment and order change are not taken into consideration because the newly added branch is already on the existing point-to-multipoint routes.

With the above modified ordering and backtracking algorithm, vertex *B* is the newly identified best branch point when new branches *E* and *G* are added.

IV. IMPLEMENTATION

We implemented the point-to-multipoint reservation service management framework in accordance with the functional layering concepts of the telecommunications management network (TMN) [16], as applied to KT's high-speed information network (HSIN) [17]. According to the TMN functional layering concept, there are four layers: service management layer (SML), network management layer (NML), element management layer (EML) and network element (NE). To implement the proposed point-to-multipoint reservation service management framework, we focus only on the service and network management layers. Our implementation model is illustrated in Fig. 9.

From the perspective of SML, there are five service management components: provisioning server (PS), integrated services and network manager (ISNM), subscriber data manipulation server (SDMS), SDMS Web server and Help Desk. All of the system components of SML and NML are implemented by CORBA objects and are communicated via the CORBA IIOP (IONA Orbix) [18]-[22].

The provisioning server maintains the network facility data gathered from network elements using the simple network management protocol. The ISNM plays a key role in network planning using the gathered network facility data provided by the server and the faults and performance data gathered from NMSs. It receives notice of all kinds of abnormalities generated by the NMSs and requests analysis or repairs from the HelpDesk system. The subscriber data manipulation server maintains subscriber information, such as service type and service period, and plays the roles of service subscription and withdrawal according to the subscriber's requests. The HelpDesk diagnoses any portion of the network and connections using Continuity Check and Loop back. It also receives the trouble ticket (TT) generated by the ISNM to repair any malfunctioning part of the network.

The ATM NMS, on the other hand, consists of four system managers: configuration, connection, fault, and performance (*ConfMgr*, *ConnMgr*, *FaultMgr* and *PerfMgr*). The *ConfMgr* maintains the network topology provided by the ISNM and controls the status of network topology. The *ConnMgr* maintains the routing information base embedding the BATT, which mirrors the network topology maintained by the *ConfMgr*. In addition, *ConnMgr* takes on the roles of setting up, releasing, and modifying the point-to-multipoint PVCs for

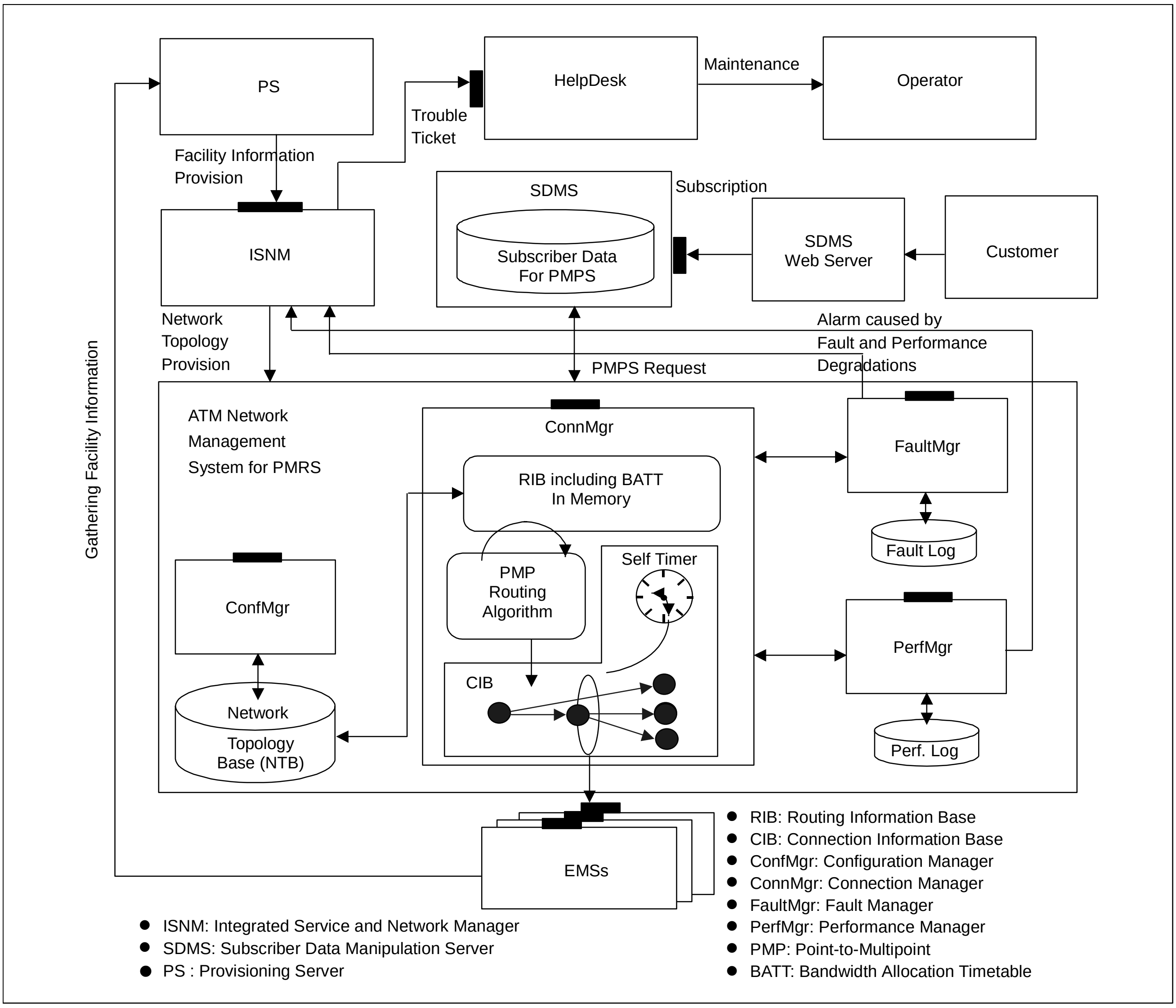


Fig. 9. Implementation model.

supporting PMRS and carries the proposed point-to-multipoint routing algorithm.

ConnMgr maintains the network topology as a routing information base, which is used to find the best point-to-multipoint route. The identified point-to-multipoint route is maintained in the connectivity information base. The *FaultMgr* detects network abnormalities and reports them to the service level management components of the ISNM. The *PerfMgr* gathers network performance metrics, such as the transmission convergence (TC) adaptor, cell level protocol, traffic load, and usage parameter control/network parameter control (UPC/NPC) every 15 minutes [12]. The ISNM periodically gathers the performance data from the *PerfMgr* of the ATM NMS and analyzes them to re-configure each network.

The procedure to provide the point-to-multipoint reservation service is as follows.

The SDMS receives a request for point-to-multipoint reservation service from a subscriber by computer, phone or fax. The SDMS asks the ATM NMS if it can support the requested subscription considering network resource utilization and the point-to-multipoint route. The ATM NMS calculates the equivalent bandwidth with the requested ATM traffic descriptors, such as peak cell rate (PCR), sustainable cell rate (SCR), maximum burst size (MBS), and cell delay variation (CDV), and then finds the best point-to-multipoint route from the named source to the destinations with the calculated equivalent bandwidth. It also adjusts and adds the BATTs of links traversing the selected point-to-multipoint route. The identified point-to-multipoint route is then stored in the connectivity information base with the service activation and

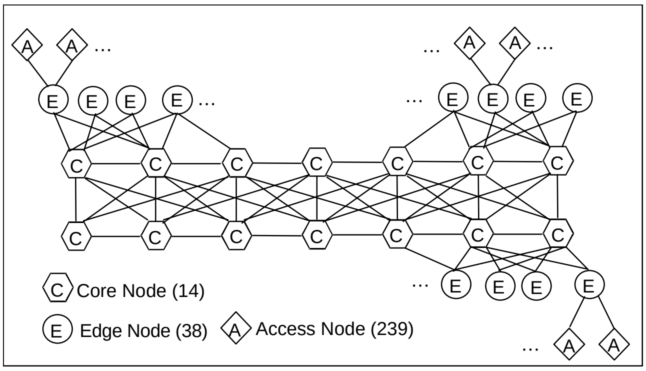


Fig. 10. High-speed information network (HSIN) topology.

deactivation time. As a result of the point-to-multipoint routing, the ATM NMS advises the SDMS whether the subscription request can be accommodated. The SDMS notifies the subscriber of the acceptance or rejection of the subscription via computer, fax, or phone.

With an internal self-timer, the ATM NMS controls the element management systems (EMSs) that belong to the point-to-multipoint route found in step 3 to set up a cross connection prior to the 30-minute real service activation time for the timely provision of the PMRS, taking into account the EMS manipulation overhead.

The service management entity (SME) in Fig. 1 is composed of service and network management systems. The service management system of the SME in Fig. 1 corresponds to the provisioning server (PS), HelpDesk, ISNM, SDMS, and SDMS Web Server in Fig. 9. On the other hand, the network management system of the SME in Fig. 1 corresponds to the ATM network management system for the PMRS composed of *ConfMgr, ConnMgr*, *FaultMgr*, and *PerfMgr* in Fig. 9.

We evaluated the performance of a real PMRS under KT's high-speed information network (HSIN) [17], which is composed of 14 core nodes, 38 edge nodes, and 239 access nodes. All the core nodes are connected in a nearly full mesh with synchronous transfer mode—4 (STM-4), while the edge and access nodes are connected to the core nodes in a star topology with STM-1. There are 291 nodes and 908 links, comprising a significantly large network.

We set a minimum reservation notification time T_{Nmin} indicating the time during which a network service provider must confirm the subscriber's reservation, which includes the point-to-multipoint route selection and the manipulation of BATTs, as well as the CAC for maximizing network resource utilization. We aim to confirm the availability of the reservation service to subscribers while they are on-line. Therefore, a mechanism to minimize notification time is crucial. We used a SUN E6500 equipped with 8 CPUs and 2 G main memory to measure the notification performance of the PMRS when branches are added. We did not use simulation tools to measure the performance because to be sure of the validity of our performance measurement, it was important to use the performance of a real service provider. Fig. 11 presents the notification performance (T_N) of up to one hundred destinations. There are two major performance metrics: ordering performance and backtracking performance. In ten destinations, it took 2.13 seconds to validate whether the requested point-to-multipoint reservation service could be accommodated. In the worst case among fifty-one destinations, it took 4.48 seconds to confirm the service availability.

The ordering performance is constant under the same network topology regardless of the number of destinations. However, backtracking performance, including CAC and

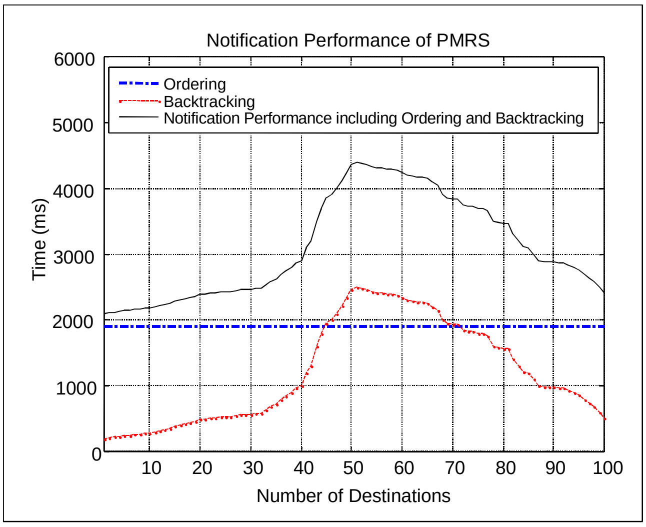


Fig. 11. Notification performance (T_N) of PMRS.

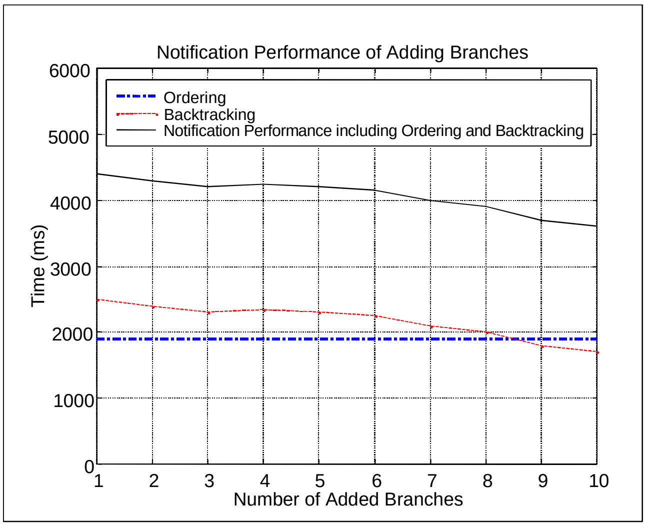


Fig. 12. Notification performance (T_N) of the adding branch.

BATT manipulation, gradually deteriorates until the number of destinations reaches fifty-one. Figure 11 shows that the same performance gradually improves in proportion to the number of destinations when it is around fifty-one. According to an analysis of subscriber trends in KT's HSIN [17], most subscribers request the point-to-multipoint reservation service for up to ten destinations. We analyzed performance under the constraint of one hundred destinations as the maximum number to validate scalability.

The ordering algorithm manifested a constant performance of 1.90 seconds under the same network topology. However, the backtracking algorithm resulted in higher performance in inverse proportion to the number of destinations when it was larger than fifty-one. This is mainly because the duplication ratio of intermediate edges and vertexes increased in proportion to the number of destinations. Our performance evaluation revealed that the proposed ordering and backtracking algorithms for providing best point-to-multipoint reservation service yielded good scalability. Because the ordering algorithm traverses the entire network topology and determines the orders of vertexes and edges, its performance was nearly constant under the same network topology. In addition, the performance of the ordering algorithm depends entirely on the complexity of the network topology. On the other hand, the performance of the backtracking algorithm depends entirely on the number of destinations.

We also evaluated the adding branch performance when the number of destinations of the existing point-to-multipoint PVCs reaches five. The performance is illustrated in Fig. 12. Figure 12 reveals that the notification performance of the adding branch is higher than that of the original point-to-multipoint route selection because it does not manipulate the BATT on the existing point-to-multipoint routes. The proposed adding branch algorithm results in higher adding branch performance in inverse proportion to the number of branches added, which is the most prominent strength of the proposed adding branch algorithm. Adding branches to the existing point-to-multipoint routes gradually resulted in improved performance in proportion to the number of branches added. It took 4.45 seconds when one branch was added to the existing point-to-multipoint routes with fifty-one destinations, while only 3.87 seconds was needed when ten branches were added.

With the performance evaluation of the proposed point-to-multipoint routing algorithm under the real network, we learned that the proposed algorithm could be applied to a real large-scale B-ISDN service network. This point-to-multipoint reservation service management framework confirms the possibility of service accommodation within 1.5 seconds when the number of destinations is below ten.

V. CONCLUDING REMARKS

This paper proposed a point-to-multipoint reservation service management framework for B-ISDN and sought to support the rapid confirmation of the subscriber's reservation and to provide best point-to-multipoint routing with efficient network resource utilization. We proposed a bandwidth allocation timetable (BATT) concept that considers the maximization of network resource utilization and a high-performance point-to-multipoint routing algorithm consisting of ordering and backtracking procedures. In addition, we proposed an algorithm for adding extra branches to the existing point-to-multipoint route.

The major concerns of the service provider and subscriber in supporting the point-to-multipoint reservation service involve rapidly confirming the acceptance of the subscriber's request and providing timely service. We implemented the proposed framework using CORBA and IONA Orbix and took into account the distributed nature of large-scale telecommunications service and network management systems. We showed that the proposed ordering and backtracking algorithms could be applied to large-scale ATM networks by analyzing the performance of the time of the point-to-multipoint route selection and the addition of extra destinations to an existing point-to-multipoint route. Our scheme was able to confirm the admissibility of the subscriber's requests within 4.03 seconds when the destination was below ten, with BATT manipulation, CAC, and best route selection included. KT provided the point-to-multipoint reservation service using the proposed framework under the HSIN [17] and had no complaints from the subscribers.

The ordering algorithm manifested a constant performance of 1.90 seconds under the same network topology. However, the backtracking algorithm resulted in higher performance in inverse proportion to the number of destinations when it was larger than fifty-one. This is mainly because the duplication ratio of intermediate edges and vertexes increased in proportion to the number of destinations. Our performance evaluation revealed that the proposed ordering and backtracking algorithms for providing best point-to-multipoint reservation service yielded good scalability. Adding branches to the existing point-to-multipoint routes gradually resulted in improved performance in proportion to the number of branches added. It took 4.45 seconds when one branch was added to the existing point-to-multipoint routes with fifty-one destinations, while only 3.87 seconds was needed when ten branches were added. A future study will focus on developing a more elegant bandwidth management mechanism for reservation service that will minimize the size of the bandwidth allocation timetables.

REFERENCES

[1] ITU-T Recommendation I.371, "Traffic Control and Congestion Control in B-ISDN," Mar. 1993.

[2] K.I. Sato and I. Tokizawa, "Flexible Asynchronous Transfer Mode Networks Utilizing Virtual Paths," *GLOBCOMM'90*, 1990, pp. 831-838.

[3] K.I. Sato and I. Tokizawa, "Broad-Band ATM Network Architecture Based on Virtual Paths," *IEEE Trans. Comm.*, vol. 38, 1990, pp. 1212-1222.

[4] Eric W.M. Wong, Andy K.M. Chan, Sammy Chan, and King-Tim Ko, "Bandwidth Allocation for Virtual Paths in ATM Networks with Dynamic Routing," *IEICE Trans. Comm.*, vol. E83-B, no. 3, Mar. 2000.

[5] Daniel W.K. Hong, D.S. Yun, "Distributed Connection Management Architecture for Best PVC Provisioning on Hierarchical ATM Transport Network," *DSOM'99*, Oct. 1999, pp. 63-75.

[6] ITU-T Recommendation I.131, "Overall Network Aspects and functions – Network Functional Principles," Sept. 1997.

[7] ITU-T Recommendation I.311, "B-ISDN General Network Aspects," Mar. 1993.

[8] Gyu Myoung Lee, Jun Kyun Choi, Mun Kee Choi, Man Seop Lee, and Sang-Gug Jong, "Flow-Based Admission Control Algorithm in the DiffServ-Aware ATM-Based MPLS Network," *ETRI J.*, vol. 24, no. 1, Feb. 2002, pp. 43-55.

[9] Jong Hyup Lee, "Design and Configuration of Configurable ATM Networks with Unreliable Links," *ETRI J.*, vol. 21, no. 4, Dec. 1999, pp. 9-22.

[10] Jun Kyun Choi, Soo Hyung Kim, Nam Kim, Sung Won Sohn, and Mun Kee Choi, "Performance Benefits of Virtual Path Tunneling for Control and Management Flows in the Broadband ATM Networks," *ETRI J.*, vol. 21, no. 4, Dec. 1999, pp. 23-28.

[11] ITU-T Recommendation E.164, "The International Public Telecommunication Numbering Plan," May 1997.

[12] ATM Forum Technical Committee, Traffic Management Specification Versions 4.0, at-nm-0056.000, Apr. 1996.

[13] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," IETF Internet RFC2011, Sept. 1997.

[14] Myung-Ki Shin, Yong-Jin Kim, Ki-Shik Park, and Sang-Ha Kim, "Explicit Multicast Extension (Xcast+) for Efficient Multicast Packet Delivery," *ETRI J.*, vol. 23, no. 4, Dec. 2001, pp. 202-204.

[15] Seok-Joo Koh, Jong-Hwa Yi, Jin-Ho Hahm, Byoung-Moon Chin, and Chee-Hang Park, "Minimizing Cost and Delay in Shared Multicast Trees," *ETRI J.*, vol. 22, no. 1, Mar. 2000, pp. 30-37.

[16] ITU-T Recommendation M.3100, "Generic Network Information Model," July 1995.

[17] Daniel W.K. Hong, "The Distributed Network Management Architecture for High-Speed Information Network," *KT Technical Document*, Apr. 2001.

[18] IONA Technologies PLC, Orbix Programming Guide – Release 2.0, Nov. 1995.

[19] OMG, CORBAServices: Common Object Service Specification, Mar. 31, 1995.

[20] OMG, The Common Object Request Broker: Architecture and Specification, July 1995.

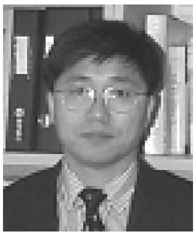
[21] IONA Technologies PLC, OrbixNotification Programmer's Guide and Reference, Dec. 1998.

[22] IONA Technologies PLC, OrbixWeb Programmer's Guide and Reference, May 1999.

[23] ITU-T Recommendation G.805, "Generic Function Architecture Of Transport Networks," Nov. 1995.

[24] ATM Forum Technical Committee, Private Network-Network Interface Specification Version 1.0, Mar. 1996.

Daniel Won-Kyu Hong received his BS and MS degrees in computer science from Hannam and Konkuk Universities, Korea, in 1987 and 1989, respectively. In 1993 he joined KT, where he worked on a TINA-C related Service and Network Management Project as a Member of Technical Staff. Since 1996, he has been involved in the project of the design and implementation of a High-Speed Internet Service and Network Management System as a Senior Member of Technical Staff. His research interests include ATM and IP services and network management architecture in a distributed processing environment, fault-tolerant management for distributed components in the inter-domain and heterogeneous networks, active network management, service level agreement (SLA), and policy-based network management. He is a member of IEEE, IEICE and Korean Institute of Communication Sciences (KICS).



Choong Seon Hong received his BS and MS degrees in electronics engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on N-ISDN and Broadband Networks as a Member of Technical Staff. In September 1993, he joined Keio University, Japan. He received the PhD degree at Keio University in March 1997. He worked for the Telecommunications Network Lab, KT as a Senior Member of Technical Staff and as the Director of the Networking Research Team until August 1999. Since September 1999, he has worked as a Professor of the School of Electronics and Information, Kyung Hee University. His research interests include the service and network management architecture for a distributed processing environment, fault-tolerant management for distributed components and mobility services between heterogeneous networks. He is a member of IEEE, IPSJ, and Korean Institute of Communication Sciences (KICS).

Dong Sik Yun received his BS in electrical and electronic engineering from Hankook Aviation College and his MS in electrical and electronic engineering from Korea Advanced Institute Science and Technology (KAIST), Korea in 1986 and 1988, respectively. In 1983 he joined KT, where he worked on an N-ISDN and B-ISDN field trial project. From 1993 to 1994, he researched TINA-C service management architecture as a Visiting Researcher of core-team member at NJ, USA. From 1998 to 2001, he was a Director of development for KT broadband multi-service (ATM, FR, and IP) and the network control and management system. Now, he is a Director and Coordinator of the project for design and development of KT's integrated service/network management system. His research interests include ATM, ADSL and IP services and network management architecture in a distributed processing environment, fault-tolerant management for distributed components in the inter-domain and heterogeneous networks, wireless internet access technology, and policy-based network management.

Jae Hyoung Yoo received his BS, MS and PhD degrees in Electronic Engineering Department of Yonsei University, Korea in 1983, 1985 and 1999, respectively. In 1986 he joined KT and worked for the standardization of TMN until 1989. From 1990 to 1999, he was a Project Manager of a group which developed TMN based NMS's covering the local, toll and international switched network. From 2000 to 2001, he was a Visiting Researcher to NTT Service Integration Labs, where he studied dynamic routing algorithms and MPLS traffic engineering. He works now as a Project Manager and leads a group developing ATM and xDSL network management systems. His research interests include ATM, IP services and network management architecture for the next generation network. He is a member of TMForum World Ordering team.

Woo-Sung Kim received his BS, in electronic engineering from Yonsei University, Korea in 1981, his MS, PhD in electrical and electronic engineering from KAIST (Korea Advanced Institute of Science and Technology) in 1983 and 1991, respectively. In 1983 he joined KT, where he worked on Directory Assistance project. He worked on M10CN exchange and ISDN during 1985-1987. From 1991 to 1996, he was a Project Manager in the development for TRUMAN (TRUnk MANagement) covering total national based trunk management including facility management, circuit provisioning, and planning. From 1997 to 2000 he worked on access network management and service order as the Team Leader in charge of the projects. From 2000, he researched a business model and XML in e-commerce network management for one year at BT (British Telecom). He has been involved in a project of IP and data network management from 2001. His research interests include ATM and IP service and network management architecture in a distributed processing environment, fault-tolerant management for distributed component in the inter-domain and heterogeneous networks, active network management, service level agreement (SLA), and policy-based network management.