Look-ahead Type Detour Path Management Methods

Yu Miyoshi†*, Tatsuyuki Kimura, and Yoshihiro Otsuka*

Abstract

We describe a "look-ahead type" detour path specification method that can specify in advance detour paths to be used when a failure occurs in a Next Generation Network (NGN) based on Internet protocol (IP) technology. A resource management server (RMS) specifies service paths using weight information of interior gateway protocols (IGPs) and specifies detour paths by assuming failure points. This method overcomes the problem of the previous method, which required a huge amount of calculation as the network scale increased because it stored all of the detour path information in a database and the number of assumed fault patterns is huge. By limiting the detour paths calculated in advance to primary ones and by calculating secondary and lower-level detour paths later, after monitoring the network situation during service operation, our new method can perform the initial setup with much less calculation than before, which enables services to be started more quickly. The effectiveness of our method was demonstrated by simulation.

1. Introduction

In the near future, many telecommunication companies in the world will be migrating to the Next Generation Network (NGN) based on Internet protocol (IP) technology. In the NGN, various services will need path information for operation functions such as service assurance and billing. A function for managing detour paths when a network element fails is important.

On the other hand, high-availability technologies for the transport network are being rapidly developed by hardware vendors. For example, fast-convergence technology that rapidly rebuilds detour paths can prevent interruptions to service operations. Though the transport networks can converge quickly, operation support systems and business support systems take a certain amount of time to calculate detour paths. To avoid discrepancies between actual network conditions and the information used for management, support systems should calculate detour paths before failures occur. However, pre-calculation requires an enormous amount of calculation and absurd assumptions about the failure points when the network scale is large.

This paper describes a new detour path calculation method that does not need as much calculation and that avoids unrealistic assumptions by using the monitoring functions of network management systems. After briefly outlining the video streaming services to which we applied our method, it reviews the previous path management method. Then, it explains our "look-ahead type" detour path management method. The method's usefulness and implementability are evaluated by simulating the calculation amount.

2. High-quality video streaming services

To offer high-quality video streaming services in an IP network, it is effective to use a bandwidth broker [1]. The bandwidth broker manages bandwidths of service paths by controlling a customer's access according to the availability of network resources. For the bandwidth broker managing high-quality services, we are developing a resource management server (RMS) that comprehends the network topology.

In a conventional IP network, a bandwidth broker

† NTT Network Service Systems Laboratories
Musashino-shi, 180-8585 Japan
E-mail: miyoshi.yu@lab.ntt.co.jp

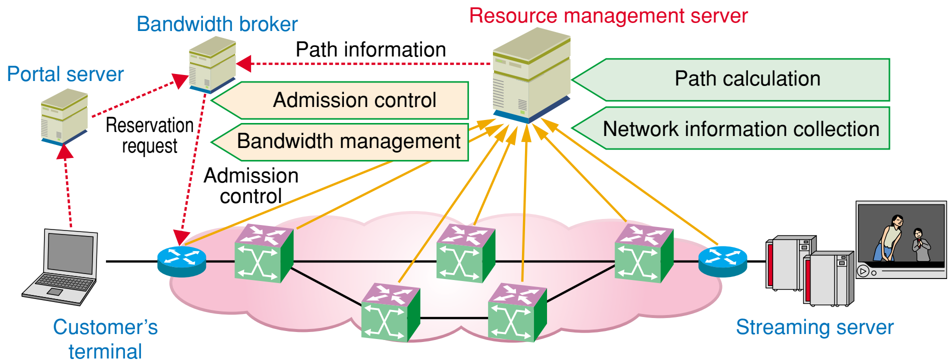

Fig. 1. Resource management server in the bandwidth broker model.

can provide video contents to customers with high quality achieved by comprehending the resource situation and controlling access to content. In NTT Laboratories, we have been investigating how to make high-quality video streaming services using the bandwidth broker model [1]. In this section, we describe the bandwidth broker model using an RMS.

First, we describe a service whose quality is suitable for live distribution or videoconferences where the time of use is fixed beforehand. We call it the quality reservation service. In this service, customers register the required bandwidth via the graphical user interface of the RMS when they want to use the network resources. The bandwidth broker checks the available bandwidth of the service path at that time and reserves it if the reservation is approved. As a result, the system can achieve high quality by performing priority control at the target time slot for the desired service. If customer demands exceed the total amount of bandwidth available, it is possible to offer alternative times and perform the necessary re-scheduling in advance. In a video-on-demand service, the demands come immediately from customers, so a reservation service of this type is not applicable. For a video-on-demand service, it is effective to provide a guarantee of quality based on an understanding of the consumption of each resource in real time and access control judgment according to the network resource situation.

In these services, a bandwidth broker needs to process a customer's access in real time. However, the burden of the path calculation on a system is large in a large-scale network. To balance the load, it will be necessary for the RMS that prepares path and resource information to be separated from the bandwidth broker that handles a customer's access request (**Fig. 1**).

3. Path management method on RMS

This section describes methods for specifying service paths (**Fig. 2**). To manage them, the RMS must specify all the links from the edge routers that connect to the server to the edge routers that connect to user nodes. If the service is interactive like a videophone, then the RMS must specify the path from the edge router connected to the caller to the edge router connected to the destination user.

First, the RMS gets the weight (cost information used by the interior gateway protocol (IGP)) and IP addresses by SNMP (simple network management protocol) or Telnet. Next, it specifies the relationships among routers using IP addresses and subnet masks. As a result, it can discover the network topology and then specify all service paths between two arbitrary points. Each service path is labeled with the minimum weights between a pair of points (shortest path). In addition, a Dijkstra algorithm is used as the shortest path selection algorithm as well as OSPF (open shortest path first), which is a type of IGP. An RMS extracts the shortest paths between every pair of nodes and stores them all.

Next, we describe the detour path calculation procedure. The RMS assumes that a certain router or link has failed and calculates service paths in the topology that bypass the assumed failure point. These paths are managed as primary detour paths. This procedure is repeated for each router or link in turn. In addition to a single-point failure, the RMS also calculates detour paths in the case of multiple router/link failures. This procedure is repeated for all shortest paths, but if a selected path produces the same results, the path is not stored.

If a primary detour path breaks down, its detour path is defined as a secondary detour path. Moreover,

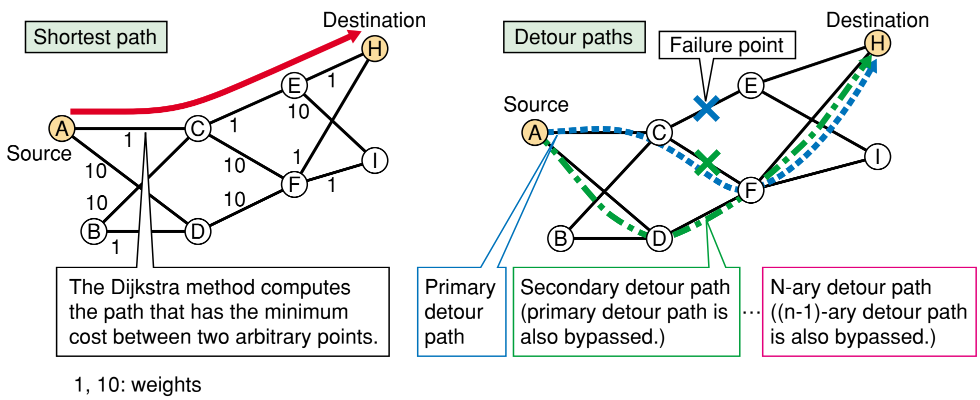

Fig. 2. Path specification.

if a secondary detour path breaks down, its detour path is defined as a tertiary detour path. If this procedure is repeated, all paths (n-ary detour paths) will be extracted. However, in a large-scale network, since the number of calculations increases greatly, the long time required for service provisioning is a problem.

4. Look-ahead type detour path management

Our "look-ahead type" detour path specification method can reduce the number of detour path calculations needed to maintain high-speed switching to the detour paths in a bandwidth-broker-type service when failures occur. Here, we compare its flow with that of our previous specification method (**Fig. 3**).

In the previous method, a bandwidth broker cannot start service operations until the RMS has finished its calculations (Fig. 3(a)). The calculation time increases with the network scale and could never end, so even though the path information is incomplete, the RMS must interrupt the calculation at some stage in order to enable the service to be started. Furthermore, there is too much path information and this makes the access load heavy. We think that the main factor is the reference speed of the bandwidth broker's load. On the other hand, when a failure occurs, the RMS is idle except for reporting SNMP traps (failure information)(Fig. 3(b)). The previous method is not realistic for commercial service operation.

The procedure of the "look-ahead type" detour path specification method is shown in **Fig. 4**. This method utilizes the advantage that the hardware of the RMS and the bandwidth broker are separated; that is, tasks are assigned to two servers. In the service preparation stage, when an RMS finishes calculating the shortest path and the primary detour paths, it transmits the path and bandwidth information to a bandwidth broker which starts service operation. This path information contains the detour paths necessary to handle the first failure to occur, and the bandwidth broker can maintain services by switching to detour path resources when a failure occurs.

Theoretically, an RMS could also calculate secondary and lower-level detour paths while assuming that the primary detour paths are broken. However, in our method, these detour paths are not calculated before a service starts (only the shortest paths and primary detour paths are calculated) because they are not used at the service start time and calculating them would require more effort than calculating primary paths since there are more failure patterns. Since the amount of calculation is limited compared with calculating n-ary detour paths in the service preparation stage, this method greatly reduces the time to service startup. However, we think that the RMS cannot provide suitable QoS with only primary detour paths during long-term operation because the probability of serious failure occurring increases with time, so more-than-secondary detour paths are likely to be required for long-term service operation.

In our "look-ahead type" detour path specification method, after the RMS has transmitted path information, it starts calculating lower-level detour paths in the background considering the present network state, which does not include any failures yet, by monitoring the network. After providing shortest paths and primary detour paths, the RMS continuously monitors the network (Fig. 4(a)). If a network failure is detected, the RMS reports the failure point to the bandwidth broker (Fig. 4(b)). The bandwidth bro-

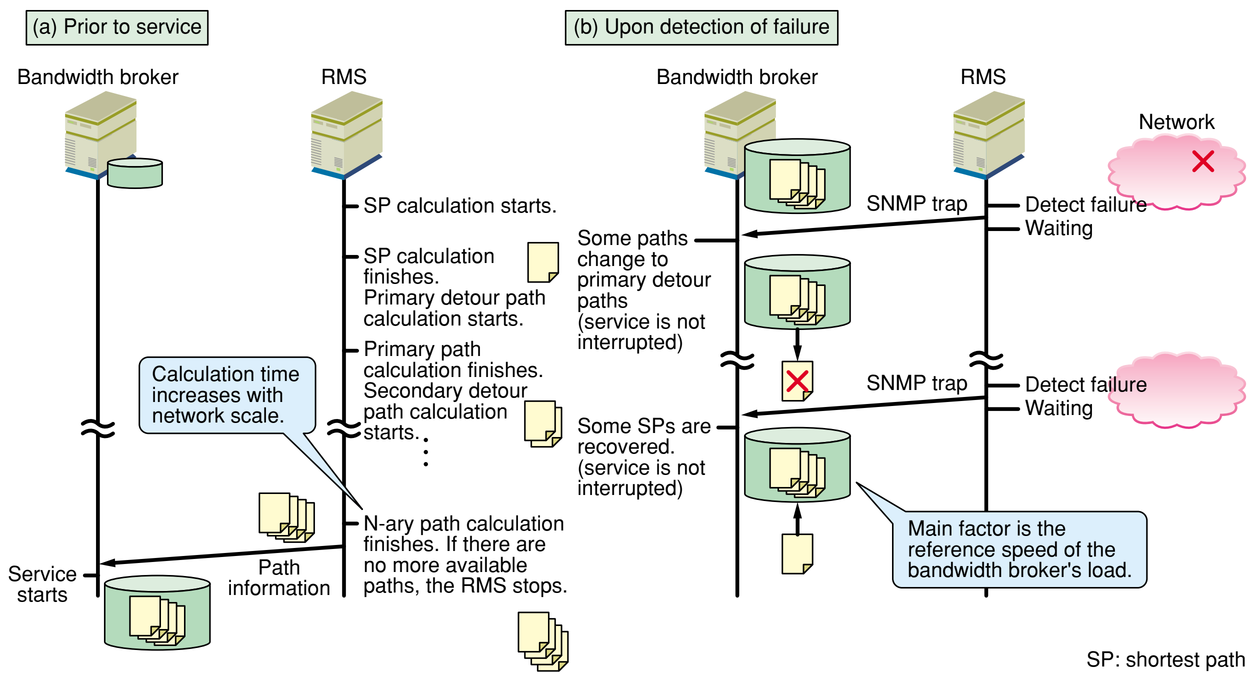

Fig. 3. Conventional detour path specification.

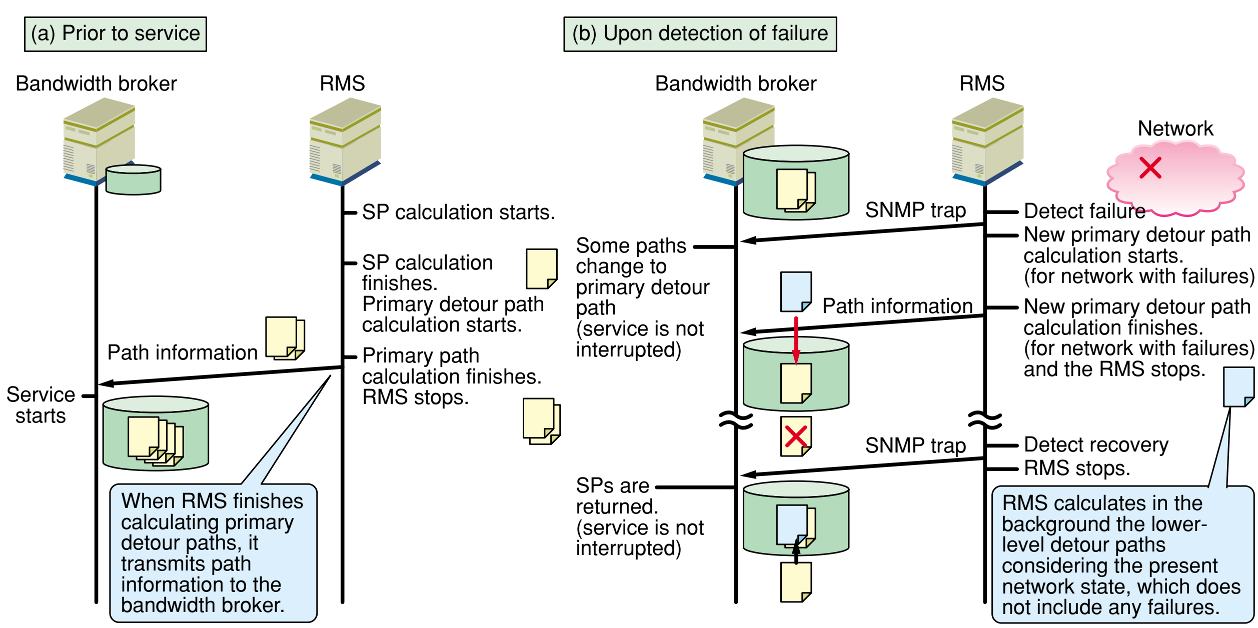

Fig. 4. “Look-ahead type” detour path specification.

ker immediately switches managed resources to detour paths without interrupting the service because it has already stored necessary paths among the primary detour paths. After reporting to the bandwidth broker, the RMS starts calculating new primary detour paths based on the network topology existing after the failure. The new primary detour paths are ones that would have been included among the secondary detour paths obtained prior to the service start if such a calculation had been performed. However, less calculation is required to obtain the new primary detour paths than to obtain all possible secondary

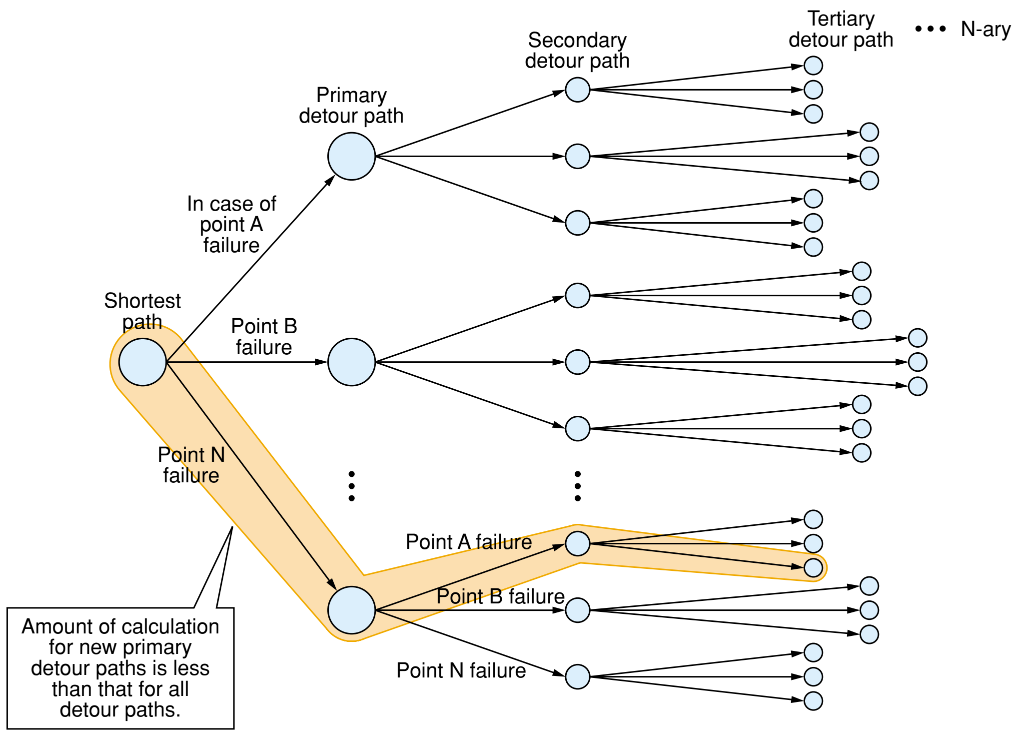

Fig. 5. Limited calculations.

detour paths before the service starts (**Fig. 5**).

When there are two or more failures, the network topology pattern is changed in accordance with the order of recovery. Therefore, the RMS must calculate the recovery path with primary detour paths, and the paths are reported to the bandwidth broker. The RMS can achieve a result equivalent to calculating the more-than-secondary detour paths by calculating primary detour paths after a failure.

5. Simulation

In this section, we evaluate the "look-ahead type" detour path specification method.

5.1 Software module

Each function of the RMS has been developed as a software module that operates on a service management platform called a service resource agent (SRA) [2] (**Fig. 6**). An SRA has a general-purpose interface to connect network management systems and network elements. Using this interface, it can collect the data it needs to perform service management by SNMP or Telnet. Moreover, the database records are designed so that one service path can be managed as one object. We were able to use these functions without changing the SRA platform and implement an RMS on an SRA as only one function module that calculates shortest paths and detour paths.

5.2 Network model

The model network topology is shown in **Fig. 7**. In this network, OSPF areas are directly connected to area 0. We assume that the basic topology in each area is a tree type. However, some paths are duplex, and detour paths exist for service redundancy. We also assume a simple topology for area 0. The two core routers are connected to each area. We should assume that the number of customers will increase and that the network topology will be changed frequently. Therefore, we verified our method in networks of various compositions having from 100 to 2000 edge routers. We think that nationwide networks can be constructed with 2000 edge routers.

5.3 Simulation results

The number of paths and the number of specified paths are shown in **Table 1**. Dijkstra calculation accounts for a large proportion of the path specification processing. Detour path specification performed by assuming failure patterns is not very efficient because of reachability loss and re-extraction of the same paths. However, the number of calculations is less than the number of paths because RMS specifies

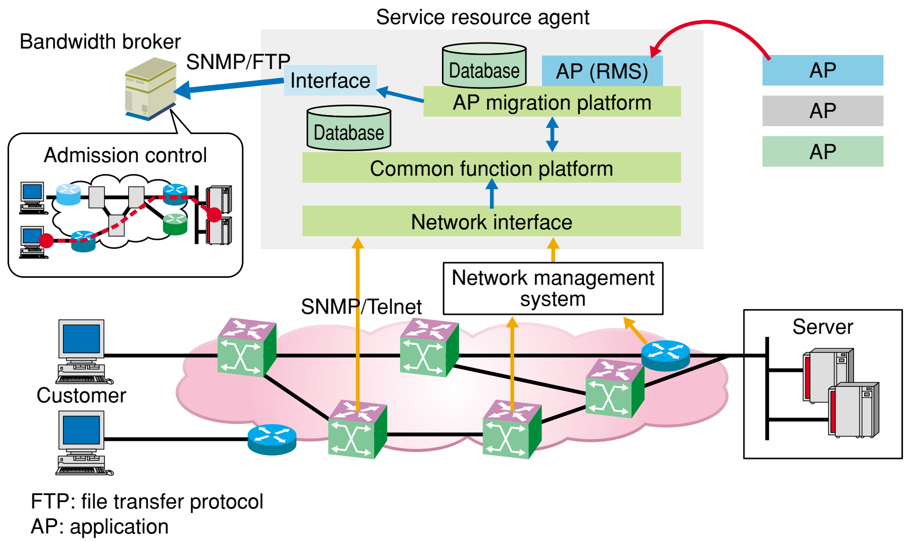

Fig. 6. Resource management server on a service resource agent.

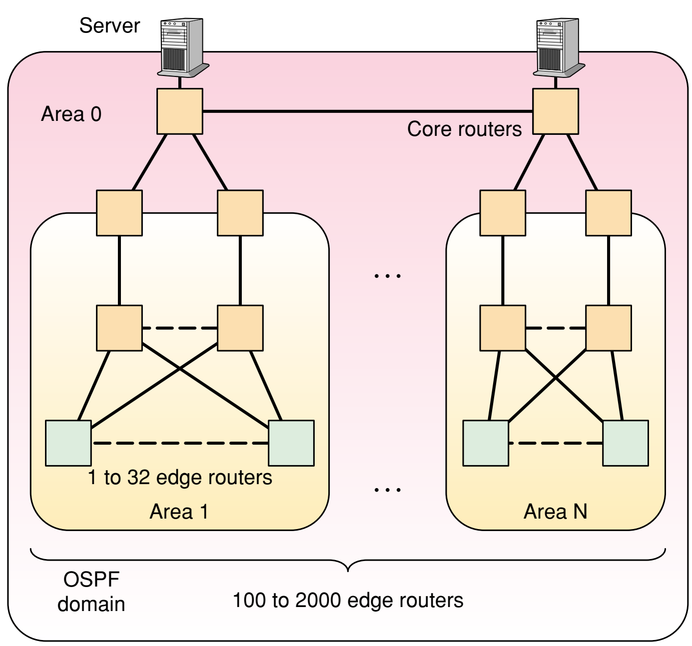

Fig. 7. Network model.

paths inside the OSPF area first, then connects the paths between areas, and finally determines an end-to-end path. The simulation showed that when calculations were performed using a general-purpose personal computer (CPU: 2.5-GHz Pentium 4), the calculation time for deriving about 4 million shortest paths in a 2000-node network was about 60 seconds. These results show that the Dijkstra algorithm is fast enough for service. Moreover, it took 11 minutes to calculate the 15 million primary detour paths. Thus, there is no serious influence on the start of service operation. However, it will probably be difficult to prepare detour paths beforehand if the “look-ahead type” method is not used because the calculation time for extracting all tertiary paths was more than 3 days. Specifying beyond tertiary detour paths takes even longer. When the “look-ahead type” method is used, the service can start after 11 minutes.

Table 1. Simulation results.

Number of Dijkstra calculations

Parentheses indicate the number of specified paths.

Edge routers	Shortest paths	Primary detour paths	Secondary detour paths	Tertiary detour paths	…
100	2956 (9900)	11,824 (31,460)	47,296 (55,730)	281,411 (356,390)	…
1000	35,240 (999,900)	140,960 (3,909,560)	563,840 (6,848,930)	3,354,848 (20,431,590)	…
2000	79,622 (3,998,000)	318,488 (15,818,520)	1,273,952 (27,696,860)	7,580,014 (82,850,380)	…
Processing time	60 seconds	11 minutes	6 hours	3 days	

It took 11 minutes to calculate the 15 million primary detour paths.

6. Conclusion

The previous detour path management method had a problem in that the RMS had to specify detour paths and inform the bandwidth broker, but the number of assumed failure patterns needed to calculate all detour paths is huge. Our "look-ahead type" detour path management method calculates only a limited set of paths by monitoring the network situation during service operation. As a result, the RMS can provide detour paths in advance without a massive amount of calculation. Simulations of the calculation amount showed the usefulness and implementability of our method. Future work should include verifying many things by changing the network model conditions. We also intend to concentrate on developing advanced services using the path information stored in an RMS.

References

[1] A. Masuda, T. Hironaka, M. Yaguchi, M. Shima, and A. Kuronaka, "Prototype implementation of a bandwidth agent for content delivery service," Technical Proceedings World Telecommunications Congress (WTC-ISS), 2002.

[2] Y. Miyoshi and T. Kimura, "Service Resource Management Architecture," NTT Review, Vol. 14, No. 3, pp. 28-34, May 2002.

[3] Y. Miyoshi, T. Kimura, Y. Otsuka, Y. Fujita, S. Majima, and K. Suda, "An Implementation of Service Resource Management Architecture," Technical Proceedings World Telecommunications Congress (WTC-ISS), 2002.

Yu Miyoshi

Network Software Service Project, NTT Network Service Systems Laboratories.

He received the B.E. and M.E. degrees in electronics, information, and communication engineering from Waseda University, Tokyo in 1998 and 2000, respectively. In 2000, he joined NTT Network Service Systems Laboratories, Tokyo, where he was engaged in R&D of network and service operation systems. He is now studying automation settings for network elements. He received the 2004 IEICE Young Researchers' Award. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and the Information Processing Society of Japan.

Tatsuyuki Kimura

Senior Research Engineer, Network Software Service Project, NTT Network Service Systems Laboratories.

He received the B.E. degree in mechanical engineering from Keio University, Tokyo in 1990. He joined NTT in 1990. Since then, he has mainly been engaged in research on transmission network management. He is currently working on the modeling of network management functions and multi-layered network management systems. He is a member of IEICE.

Yoshihiro Otsuka

Senior Research Engineer, Supervisor, Network Software Service Project, NTT Network Service Systems Laboratories.

He received the B.E. degree in electrical and electronic engineering from Kanazawa University, Ishikawa in 1983 and the M.E. degree in physical electronics from Tokyo Institute of Technology, Tokyo in 1985. Since joining NTT in 1985, he has mainly been engaged in research on broadband switching systems and network management systems. He is currently working on the modeling of network management functions and multi-layered network management systems. He is a member of IEICE.