論文 97-22-4-21

An Integration Architecture for the ATM Customer Network Management

Jong-Tae Park* *Regular Member*

ATM 고객망관리를 위한 통합 구조에 대한 연구

正會員 박 종 태*

ABSTRACT

As enterprises use ATM networks for their private networks and as these private networks use public ATM networks for wide area communication, the need for the customers to be able to manage both private and public networks. Currently, some standardization work is being done towards providing this capability to customers. In this paper, we propose a new customer network management (CNM) system architecture for the management of both ATM a private network and a public network in a uniform way. The particular features of the proposed architecture lies in the efficient support of the complex hierarchial TMN manager-agent relationships at M3 and M4 interfaces, and the support of SNMP and CMIP integration which is necessary for the implementation of a CNM system. The TMN hierarchical many-to-many manager-agent relationships are realized by the utilization of CORBA-Based SMK (Shared Management Knowledge) implementation. We have also implemented the prototype of a ATM CNM system, and measures the performance for the demonstration of the suitability of the proposed architecture.

요 약

기업의 사설망으로 ATM(Asynchronous Transfer Mode)의 채택이 증가하고, 이 사설망들이 광역 통신(wide area communication)을 위해 공중 ATM 망을 사용함에 따라, 고객은 공중망을 통하여 연결된 사설망을 효율적으로 관리하기를 원한다. 고객에게 이러한 기능을 제공하기 위한 국제 표준화가 현재 진행 중에 있다. 이 논문에서는 획일적인 방법으로 ATM 사설망과 공중망을 관리하기 위한 고객망관리(Customer Network Management: CNM) 시스템 구조를 제안한다. 제안된 구조는 M3 및 M4 인터페이스에서 TMN 계층적 관리자-대리자 관계를 효율적으로 지원하고, CNM 시스템의 구현에 필수적인 SNMP(Simple Network Management Protocol)와 CMIP

*경북대학교 전자·전기공학부

論文番號:96382-1210
接受日字:1996年 12月 10日

(Common Management Information Protocol)의 통합을 지원한다. TMN 계층적인 다대다 관리자-대리자 관계는 CORBA(Common Object Request Broker Architecture)-기반의 공유관리지식(Shared Management Knowledge)을 사용하여 실현된다. 또한, 본 논문에서는 ATM CNM 시스템의 프로토타입을 구현하고 제안한 구조의 적합성을 검증하기 위해 성능을 측정하였다.

Ⅰ. Introduction

Asynchronous Transfer Mode(ATM) has been selected as the standard technology for Broadband Integrated Services Digital Network(B-ISDN), for supporting telecommunication services and various multimedia applications. That will provide high quality services cost-effectively by means of efficient use of network resources in communication network environments independent of communication media such as voice, data, image and video. ATM has rapidly transited from a standards and prototyping concept to become the next generation switching technology for LANs and WANs. ATM networks are now available on the market and more and more enterprises are adopting them for their private local and wide area networks. There are many issues yet to be resolved before ATM becomes a mature technology. One of these issues is the operations and management of ATM networks.

As private networks need to communicate each other (due to enterprises having multiple private networks throughout the country or throughout the world), these private networks often use public network in order to exchange information between private networks. In such situations, network administrators of private networks generally would like to be able to monitor or manage their corporate network as well as public networks which their private networks use. Currently, this is a very difficult task mainly due to a couple of reasons: administrative and technical. The first reason is that public network administrators do not want their networks to be managed by private network administrators. The second reason is that there is no clean mechanism available for the private network administrators to be able to manage any portion of the public networks. Recently, however, the trend is to allow the private network administrators to manage a portion of the public network essential to the operation of custmer network management, the specific requirement of which is described in the subsequent Section II [4, 5]. Our work focuses on the technical aspect of allowing the public networks to be managed by private network administrators.

Some work has been carried by standardization bodies such as ATM Forum, ITU-T and ETSI for providing the solution [1, 2, 3, 4, 5]. These works mostly focus on defining the interfaces for ATM network management and on the specification of management information base(MIB). ATM Forum has specified five management interfaces for the exchange of management information for ATM networks. Among them, M3 is a management interface between customer network management system(CNMS) and public network management system(PNMS), and M4 is the interface supporting network element level and network level management of public ATM networks. The M3 service is provided by the public network provider via customer network management (M3) agents in the provider's network. CNM is a concept providing capabilities for network providers and their customers to exchange management information[6]. CNM services provide the customer using the provider's network with the capability to manage the portion of the public ATM networks.

Earlier work on CNM service provisioning stated the necessity of interworking between the CNM agents and public network managers [6, 7]. However, they did not provide solutions for the efficient interworking mechanisms at M3 and M4 interfaces.

Our work presented in this paper attempts to provide that solution. In particular, we propose an architecture for public network management system supporting CNM services. We are developing a public network manger which integrates CNM agents based on the M3 and M4 standard management interfaces. Our integrated network management system incorporates a CORBA-based Shared Management Knowledge (SMK) system [8, 9] which provides a distributed processing environment for the exchange of management information between managers and agents.

This paper is organized as follows. Section 2 describes standards forwarded by ITU-T and ATM Forum to provide CNM service in ATM networks, and describes previous works. Section 3 presents our proposed architecture for integrating CNM agent and public network manager. It also describes mechanisms for the efficient provision of CNM services. In Section 4, a prototype implementation carried out as a proof of concept is described in detail. The performance evaluation of the prototype is done in Section 5. Finally, a summary and possible future work are given in Section 6.

Ⅱ. CNM system requirements in ATM networks

The ITU-T recommendation X.160 [4] defines an architecture for the Customer Network Management Services for public data networks. The CNM functional architecture is composed of different function blocks that provide the functionality needed for CNM. The customer's management function provides the customer-related CNM functionality. The service provider supports the CNM function. The service boundary between the customers management function and the CNM function is defined by the CNM reference point. Management information is exchanged between the customer's management function and the CNM function through the CNM reference point.

The CNM function comprises several functional components: CNM information, access control, CNM application and mapping. The CNM functional architecture can be mapped onto the TMN(Telecommunication Management Network) functional architecture [4]. Customer's management function and the CNM function correspond to the operations system functions. The CNM reference point can be mapped onto the TMN X-reference point [6]. The CNM physical architecture is composed of customer network management system and a service provider network management system. These two management systems are connected by the M3 interface.

CNM manager communicates with CNM agent within public ATM network through the M3 interface. This communication takes place using SNMP over UDP. At the physical level, the communication may use the ATM UNI or dedicated circuit. The M3 specification is classified into two classes to allow public network providers to offer modular, incremental capabilities to meet different levels of customer's needs. The first class of M3 functions, Class I, is that a public network provider can provide information on the configuration, fault and performance management of a specific customer's portion of a public ATM network. For instance, the information may be the performance management data for a UNI link or the report of an alarm message for the loss of a UNI link. The second class of M3 functions, Class II, is that a customer can request the addition, modification or deletion of virtual connections and subscription information in a public ATM network. In a nutshell, the Class I provides monitoring capability only whereas the Class II provides controlling capability as well.

Figure 1 shows the interworking structure between CNMS and public NMS, in particular the relationship of a CNM agent and a public network manager. Recall that the public network management system is to provide either the Class I or Class II services to the customers. It is possible to provide the Class I service even if the CNM agent manages public network without interworking with the public network manager. However, it is absolutely necessary for the CNM agent to interwork with the public network manager if the Class II service is to be provided. This is mainly due to the control (add/modify/delete) capabilities associated with the Class II service provided to the customers who may potentially create problems if they are not properly mediated by the public network manager.

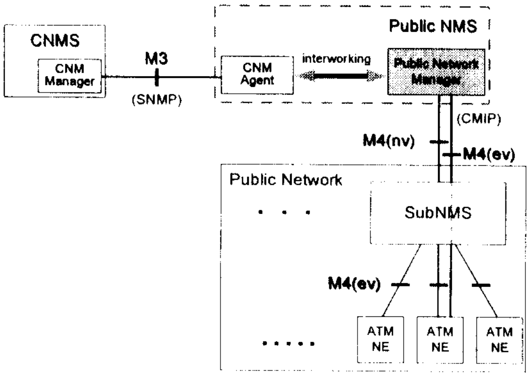
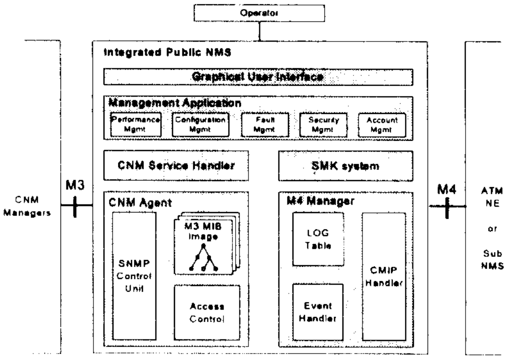


Fig. 1. Interworking between CNM agent and public network manager

Most of previous works [7, 10] proposed new information models or implementation methods to improve the performance of CNM manager and agent. However, the interworking mechanism between public network manager and CNM agent and the relationship between M3 MIB and M4 MIB in public ATM network elements have not been studied. In the next section, we present our work on a framework which includes the interworking mechanism between CNMS and PNMS as shown in Figure 1.

Ⅲ. CNM System Architecture

Figure 2 describes the proposed architecture of a public NMS which integrates a public NMS and CNM agent. The integrated public NMS provides CNM services to CNM managers. The unique feature of the architecture is that it incorporates shared management knowledge (SMK) system [8, 9] to enable distributed management of CNM and public NM systems. A public NM system may require access to several subordinate network management system, i.e., subNMSs to accomplish end-to-end management tasks. Consequently, virtual private network management requires a multi-level hierarchical manager-agent structures based on TMN interactions where there are many-to-many relationships between managers and agents. This requires efficient distributed management architecture and mechanism for managers and agents which are distributed across the network, and interconnected in a hierarchical way. This distributed management functionality is accomplished by CORBA-based SMK system as shown in Figure 2.



* SMK Shared Management Knowledge

Fig. 2. Integrated Public NMS Architecture

The major components of the integrated architecture are described in more detail below.

1. CNM Agent

The CNM agent provides CNM services to CNM managers. As shown in Figure 1, the interaction between the CNM manager and CNM agent is accomplished at the M3 interface using SNMP as the management protocol. The SNMP control unit receives requests from a CNM manager and passes them to the M4 manager through the SNMP-CMIP gateway. It also handles replies and event reports from the M4 manager and forwards them to appropriate CNM managers. The access control is used to check the permission of management requests and controls the access of management information. There may be multiple CNM managers accessing a single CNM agent, information on these CNM managers is also stored here. Since the actual MIB is located in the managed objects of the public network, M3 MIB image is used to reflect the actual MIB and is maintained by the CNM agent.

2. CNM Service Handler

CNM service handler plays an important role in the provision of CNM service. This module is composed of configuration database that maintains configuration information of specific customer's networks, management information mapping routine, and polling routine. The configuration database includes customer identifiers, and network resources for end-to-end connections between customer's remote sites such as NE's address, virtual path identifier(VPI), virtual channel identifier(VCI) and so on. When CNM Service handler receives the request from the CNM agent, it refers to this database for determining which public network resources are related to specific customer's network.

This module parses SNMP requests of CNM agent and forwards the request to M4 manager after translating the requests into recognizable forms to the M4 manager. While the management protocol for CNM is SNMP, the management protocol for managing public ATM network is CMIP. Also, when the M4 manager receives CMIP replies and event reports, those intended for CNM managers must pass through this module, converting them into those that can be handled by the CNM agent. This procedure includes the mapping between M3 and M4 managed object class, managed object attributes. We use CMIP/SNMP conversion functions described in [11].

3. M4 Manager

The M4 manager performs management interactions with public network management (M4) agents on behalf of the public network management applications as well as the CNM agent. These are two kind of management requests, one from the public NM applications, the other (i.e., CNM requests) from the CNM agent via the CNM service handler. These requests must be distinguished by the M4 manager so that the replies and event reports from M4 agents can be delivered to appropriate managers. The LOG table is used to log all the interactions that take place within the integrated manager.

4. SMK System

The exchange of management information between two function blocks in TMN requires the common view about the protocol knowledge, managenctions, managed object classes and their instances, and authorized capabilities. Such information is collectively referred to as shared management knowledge (SMK) and is defined in ITU-T M.3010 [12].

In Figure 2, there exists two processes at the SMK system: an SMK server and SMK client. The SMK server in integrated public NMS performs the functionality of providing the information on CNM agent which is related to M3 MIB, their instances and access policy to the SMK client. The SMK client in the NMS requests the SMK information to the SMK server in M4 agent. SMK Information Repository (SIR) exists in the SMK system, and it is the place for the storage of SMK information. The SMK server can, with direct access to M3 MIB image, obtain the management information without accessing the SIR. In case where the contents of M3 MIB change very often, and the size of M3 MIB is relatively small, it may be more efficient to directly access the M3 MIB to get the SMK. However, if the contents of M3 MIB are mostly static, and the size of MIB is large, it may be more efficient to store them in SIR in advance. This can be achieved by making the SMK server access the SMK information and store them in SIR when the agent process is activated. As the SMK client object requests the SMK information, the SIR is accessed for the provision of required SMK information. In this case, the M3 MIB is only accessed once each time the agent process is activated. The management of SMK information in a sense is similar to the view maintenance problem in distributed database management system. The suitable SMK management strategy may depend on the update frequency and size of M3 MIB, the complexity of M3 MIB access functionality, and of course the semantic correctness of SMK update operations [8].

5. CNM Service Mechanism

Let us consider the case where a public network provider supports M3 Class I service. When a CNM agent receives a CNM service request from a CNM manager, it converts the request into a request on the public NM application. Upon receiving this request, the M4 manager invokes management operations to M4 agent and waits for the response. These procedures may take a long time. In order to take actions in a real time, NMS periodically polls the customer-related M4 agent and update M3 MIB image. This polling is independently performed to CNM requests. When a CNM request is received, the value of M3 MIB image can be returned immediately. This may result in better performance improvement for the CNM applications. When a specific event occurs which requires modification to the M3 MIB image, NMS transfers the event to CNM agent through event handler directly. These procedures are shown in Figure 3.

The algorithm that polling process updates the M3 MIB by reading the M4 MIB is described below.

Step 1) The process invokes the M4 manager with the arguments such as the managed object identifier (OID) and instance identifier by traversing the MIB mapping table.

Step 2) The M4 manager issues a M-GET. request with this arguments to an agent.

Step 3) When M4 manager receives a M-GET. response, it parses the response and transfers it to CNM service handler. CNM service handler updates the M3 MIB. If all the M3 MIB is updated, it sleeps until the next polling time. If not, go to Step 1.

When a CNM agent receives a SNMP SET request for the modification of management information, in the case that network provider supports M3 class II service, it invokes M4 manager for the provision of update services. M4 manager, in response to the request from the CNM agent, stores service profiles in LOG table and requests M4 agents to update the related management information in ATM NE directly or via subNMS indirectly. M4 manager is responsible for modification, addition and deletion of connections and subscription information in a public ATM network

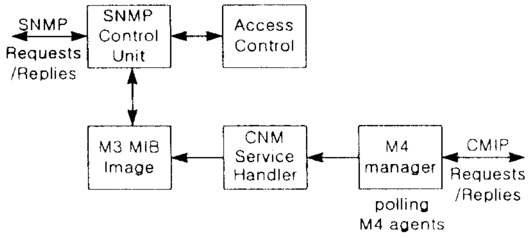


Fig. 3. Procedure to provide Class I service

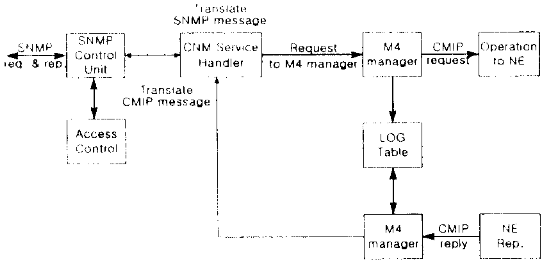


Fig. 4. Procedure to provide Class II service

according to the customer's request. The response of M4 agent is transferred to a M4 manager and then to the CNM agent based on the information stored in LOG table. These procedures in NMS are described in Figure 4.

Figure 5 shows an example of the CNMS and PNMS, NEs operations in the case that a customer wants to establish a new VP (virtual path) connection between the remote sites. At first, CNMS will generates SNMP SET message and transfers it to PNMS through M3 interface. If PNMS receives this message, it will invoke CNM service handler. CNM service handler checks the VPN configuration database to select UNIs of the two sites and related public NEs, and instructs the related NEs to make appropriate VPLs and crossconnections in turn. M4 agents performs the management functions on the managed objects at the request of the M4 manager and send back notifications to PNMS.

Event notification capability provides a customer with the ability to receive unsolicited events from public networks upon detection of abnormal conditions related to the customer premise network. Event notifications could include changes in the status of ATM UNI configuration parameters, PVC (permanent virtual circuit) configuration parameters, or access authorization failures. Event notifications are helpful to the customer for isolating failures in the customers network. Figure 6 shows the procedure of event report handling.

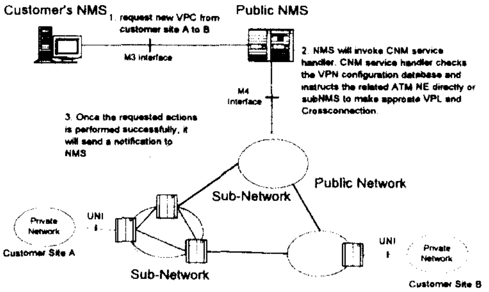

Customer's NMS
1. request new VPC from customer site A to B
Public NMS
M3 interface
2. NMS will invoke CNM service handler. CNM service handler checks the VPN configuration database and instructs the related ATM NE directly or subNMS to make approate VPL and Crossconnection.
M4 interface
3. Once the requested actions is performed successfully, it will send a notification to NMS
Public Network
Sub-Network
Private Network
UNI
Customer Site A
Sub-Network
UNI
Private Network
Customer Site B

Fig. 5. Scenario of VP connection establishment

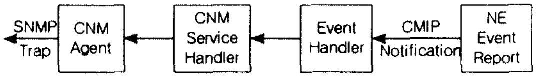


Fig. 6. Procedure of event report handling

Ⅳ. Prototype Implementation

Figure 7 shows a prototype implementation of the integrated CNM agent and M4 manager, M4 agent and SMK system. We used the OSIMIS platform which was developed at UCL [13] for implementing manager-agent interactions. It provides an environment for the development for management applications which hides the details of the underlying management service through the object-oriented application program interfaces (APIs) and allow implementers to concentrate on the intelligence to be built into management applications rather than the mechanics of management service/protocol access.

In the SMK system, the CMIS scoping in the M4 interface and SNMP in the M3 interface is not a suitable technique for the identification of complete set of MO classes, and a more generic identification mechanism was needed [8]. Thus, we used the CORBA-based approach. The SMK client object of a management application is associated with the SMK server object at the managed system through the communication facility provided by the underlying ORB (object request broker). The SMK information can be accessed by using the interface which is described in Interface Description Language (IDL). CORBA IDL and ORB implementation embedded in ORBeline [14] from PostModern Computing Technologies was used as a CORBA implementation. The communication mechanism of the CORBA ORB enables the location transparency between distributed objects to be supported, and this created the dynamic, efficient distributed processing environment.

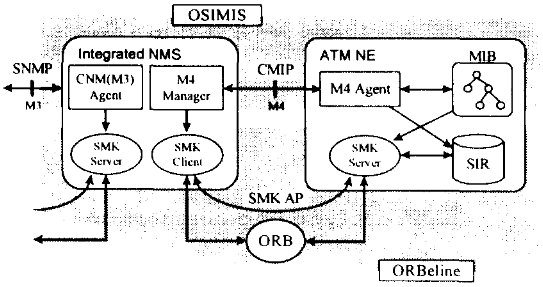


Fig. 7. Protype Implementation using OSIMIS and ORBeline

V. Performance Evaluation

In this section, we present performance results of our implementation focusing on SMK. Figure 8 illustrates the system configuration for the performance evaluation. The first host A is configured with a manager and an agent and the second host B is configured with only an agent. The manager in turn consists of a manager process and SMK client process. Both agents consist of an agent process and SMK server process. Since there does not exist any SMK system implementation (which we are aware of), it is difficult to compare our results with others. Thus, we simply present our results and describe the performance characteristics that we believe are important.

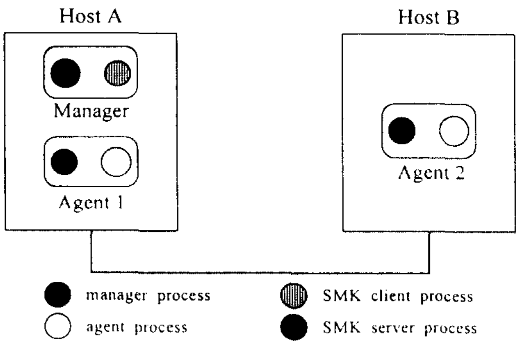


Fig. 8. The configuration of processes for the performance evaluation

Figure 10 shows the performance results of binding times between SMK client object to a SMK server object. Binding time represents the time for SMK client object to discover the desired SMK server object and connecting to it, and thus this is a very important performance metric. The times are measured in milliseconds(msec). The columns represent the number of SMK server objects and the rows represent average binding times between a SMK client objet to a SMK server object. The number of trials is 10 times. We have used a broadcasting technique for locating and binding a desired SMK server object. In order to evaluate the scalability of our solution, we have measured the binding times by doubling the number of SMK server objects repeatedly. The results shown in Figure 9 indicates the increase of binding times as the number of SMK server objects increases. However, the increase is not directly proportional to the rate of the increase.

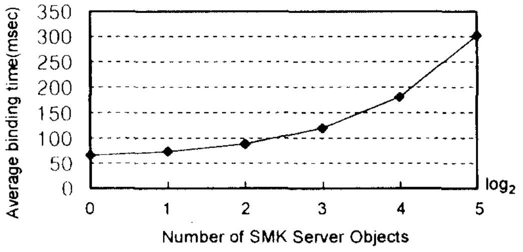


Fig. 9. Binding time between SMK objects

Our next evaluation involves the measurement of the times for SMK operations using the CORBA IDL interface and for the OSI management operations using the CMIS M-primitives. In the measurement, we have not counted the times taken to perform the requested operations since they mainly depend on the internal implementation of SMK server object and agent process as well as the platform these processes run on. The service requests used in the experiment are Get_MOClassList and Get_MOInstanceList for the ORB SMK interface and M-Get, and M-Action for the CMIS management operations. Get_MOClassList is a SMK interface parameter to get the list of MO classes in a MIB. Get_MOInstanceList is a SMK interface parameter to get the list of MO classes in a MIB.

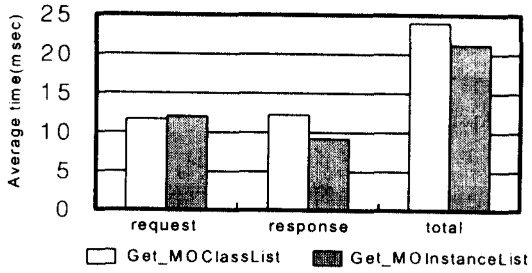


Fig. 10. Request and response time of service through ORB

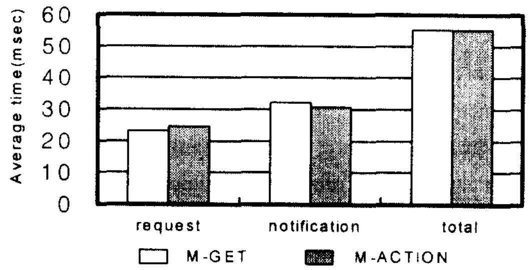


Fig. 11. Request and notification time of CMIS management service

Figure 10 shows the times measured for the SMK service requests and replies using ORB and Figure 11 illustrates the times measured for the CMIS requests and replies. The performance results as shown in these two tables indicate that the CMIS operations take two to three times longer than the ORB SMK operations. The results indicate that the overhead of using the SMK client and server objects compared to the manager-agent operations is not significant. The approach suggested in the NMF OMNIPoint uses the SMK MOs explicitly, which requires the use of manager-agent CMIS operations for SMK operations. OMNIPoint's approach would result in the performance degradation considerably since the CMIS operations take much longer. We believe our approach is superior as shown in our performance evaluation, in addition to more powerful functionality inherent to distributed object-oriented programming paradigm offered by CORBA.

Ⅵ. Conclusion

ATM network management system is absolutely necessary for providing the reliable communication services in ATM networks. The research in ATM NM has been being done by many researchers as well as by standardization organizations. We presents a new architecture for the realization of ATM network management, focusing on the efficient provision of customer's network services. The ATM NM model suggested by ATM Forum has 5 management interfaces for the exchange of management information. Each interface has related MIBs defined with different management purposes. Among these interfaces, the M3 and M4 interfaces are related to CNM service provisioning and public network management.

We have designed a customer network management system architecture which are comprised of functional models for supporting each CNM functions and public network management applications. The particular features of the proposed architecture lies in the efficient supporting of complex hierarchial TMN manager-agent relationships at M3 and M4 interfaces, and the supporting of SNMP and CMIP integration which is necessary for the implementation of M3 and M4 interfaces. The TMN hierarchical many-to-many manager-agent relationships are realized by the utilization of CORBA-Based SMK implementation. The SMK system architecture is designed, and the corresponding interfaces are specified in CORBA IDL. We have also implemented the prototype ATM CNM systems, and measures the performance for the demonstration of the suitability of the proposed architecture. The prototype implementation of our design was made by using the OSIMIS platform and ORBeline.

For the future work, we plan to investigate the mechanism of information exchange between public NMSs through the TMN X interface and interworking with CNM agents in the case where the customers PVCs traverse multiple public networks. We plan to deploy our prototype in an operational ATM testbed of Korea Telecom.

REFERENCES

1. ATM Forum, Broadband Inter Carrier Interface (B-ICI) Specification, Version 1.0, August. 1993.
2. ATM Forum M3 Specification Revision 1.03, Customer Network Management for ATM Public Network Service, 1994.
3. ATM Forum, CMIP Specification for the M4 interface, September 1995.
4. ITU-T Recommendation X.160, Architecture for Customer Network Management Service for Public Data Network, 1994.
5. ITU-T Draft Recommendation I.751, ATM Management of the Network Element View, November 1995.
6. Jacqueline Aronsheim-Grotsch, "Customer Network Management:CNM," Proceedings of IEEE/IFIP Network Operations and Management Symposium, Kyoto, Japan, April 1996, pp. 339-348.
7. Miyoshi Hanaki, "LAN/WAN Management Integration using ATM CNM Interface," Proceedings of IEEE/IFIP Network Operations and Management Symposium, Kyoto, Japan, April 1996, pp. 12-21.
8. Jong-Tae Park, Su-Ho Ha and James W. Hong, "Design and Implementation of a CORBA-Based TMN Shared Management Knowledge System," Proceedings of IEEE/IFIP Network Operations and Management Symposium, Kyoto Japan, April 1996, pp. 64-74.
9. Jong-Tae Park and James W. Hong, "Implementation and Performance of a TMN SMK System," Proceeding of IEEE/IFIP Distributed Systems: Operations and Management, La Quila, Italy, October 1996.
10. Michael Hinchliffe and Nigel Cook, "Customer Network Management and ATM Networks," Proceedings of IEEE/IFIP Network Operations and Management Symposium, Kyoto, Japan, April 1996, pp. 155-164.
11. J.T. Park, Y.W. Choi and J.D. Kim, "The Integration of OSI Network Management and TCP/IP Internet Management using SNMP," Proceedings of IEEE First International Workshop on Systems Management, Los Angeles, April 1993.
12. ITU-T Draft Recommendation M.3010, Principles of Telecommunication Management Network, 1995.
13. G. Pavlou, "The OSIMIS TMN Platform:Support for Multiple Technology Integrated Management Systems," Proceedings of the 1st RACE IS&N Conference, Paris, November 1993.
14. PostModern Computing Technologies, ORBeline 2.0 Users Guide, 1996.

박 종 태(Jong-Tae Park) 정회원

1971년~1978년:경북대학교 공학사(전자공학).

1979년~1981년:서울대학교 공학석사(전자공학).

1981년~1987년:미국 Michigan 대학 공학박사(전기전산).

1985년~1987년:미국 Michigan 대학 Center for Information Technology Integration (CITI) 연구원.

1988년~1988년:미국 AT&T Bell Lab. 연구위원

1988년~1989년:삼성전자 컴퓨터시스템 사업부 수석연구원.

1989년~현재:경북대학교 전자. 전기공학부 교수.

※주관심분야:통신망 관리 시스템 설계 및 개발, 초고속 통신망, 분산 멀티미디어 시스템 이동통신