A Design of TINA-based Performance Management Architecture

Seoung-Woo Kim*, Young-Tak Kim* *Regular Members*

ABSTRACT

In order to guarantee the user-requested quality-of-service(QoS) and keep the network utilization at maximum, it is required to manage the network performance continuously after the network installation. The performance management function should provide the useful information for the network expansion and the capacity reallocation in the future. Currently, the TINA provides the specification of the management functions of configuration management, connection management, and fault management; but the management functions of performance management and security management are not well-defined yet.

In this paper, we propose a TINA-based performance management architecture for the efficient performance management of the heterogeneous networks or NEs with TMN and SNMP management functions. And we examine the proposed architecture into the ATM network(with SNMP and TMN) monitoring. The proposed architecture is based on the distributed processing architecture and the concept of TMN performance management.

The proposed architecture have been designed and implemented in multiprocess and multithread structure.

Ⅰ. Introduction

The current telecommunication network is composed of various high-speed and broadband network technologies, such as ATM, SDH/WDM, and IP networks. Also its users request various services. In the forthcoming next generation high-speed and broadband network, the provisioning of broadband mobile multimedia services is the most important issue; while an efficient network management architecture that can manage efficiently the network resources is essential. To provide both the efficient management of the network resources and the conformance of the users various requirements, the performance management function is essential.

In order to interwork the various management functions efficiently, the distributed network management architecture such as TINA is the most appropriate. The TINA has well specified management function of configuration, connection and fault management. But the management functions for performance management and security management are not well-defined yet; most of the functions are left as further study.

In this paper, we propose a TINA-based performance management architecture, and implemented and evaluated a part of the proposed performance management functions

The rest of the paper has the following structure. We first look briefly at the performance management of TMN, which provides the underlying framework on which our approach based. We then discuss the designed performance management architecture and its functions. We finally test the performance monitoring function and evaluate our approach.

Ⅱ. Performance Management Functions in TMN

The TMN(Telecommunications Management Network) has been defined for management of the

* 영남대학교 대학원 정보통신공학과(free@infocom.ice.yeungnam.ac.kr),
논문번호 : 00209-0615, 접수일자 : 2000년 6월 15일

public telecommunication networks[1,2]. The concepts of TMN management functions are very useful and can be applied to various networks. The TMN describes performance management as follows: Performance management provides functions to evaluate and to report upon the behavior of telecommunication equipment and the effectiveness of the network or the network element[3].

The components of performance management functions should include the followings[3]:

- Performance Quality Assurance which supports decision processes that establish, as the state-of-the art expands and customer needs change, the quality measures that are appropriate to the area of performance management.
- Performance monitoring that monitors the continuous performance data of the NE.
- Performance Management Control that supports the information delivery to control the operation of the network.
- Performance Analysis that evaluates the performance level of the network entity.

The performance management components are performed with OSI SMF(System management Functions) such as the Metric Monitoring Function, Summarization Function[4,5].

Ⅲ. Design of TINA-based Performance Management Architecture

The TINA management architecture is based on the concept of the TMN management functions[6-8]. The current status of TINA does not provide well-defined performance management functions. To achieve the performance management objectives, the TINA-based performance management should include the following functions[6-8]:

- Performance Monitoring that performs the acquisition and maintenance of activity data required for analyzing the performance measurements.
- Performance Analysis that analyzes the performance related data, determining meaningful level of performance.
- Performance Control which is associated with the specifying of the managed objects under performance surveillance, and the regulating how performance events are related to performance management entities for subsequent analysis.
- Performance Tuning that reconfigures the managed object attributes in order to maintain a desired level of performance.

Fig. 1 shows the proposed TINA performance management architecture based on TMN performance management functions. The proposed performance management architecture includes the function modules explained in above, such as Performance Monitoring(PM), Performance Analysis and Control(PAC), Performance Tuning(PT)[9]. The PM module performs the workload monitoring and the summarization functionality. The workload monitoring is to collect or capture the data of observed network resources by polling or event-driven method. The PAC module executes the data analysis algorithms and delivers the information to control the operation of the network, but it does not have direct interaction

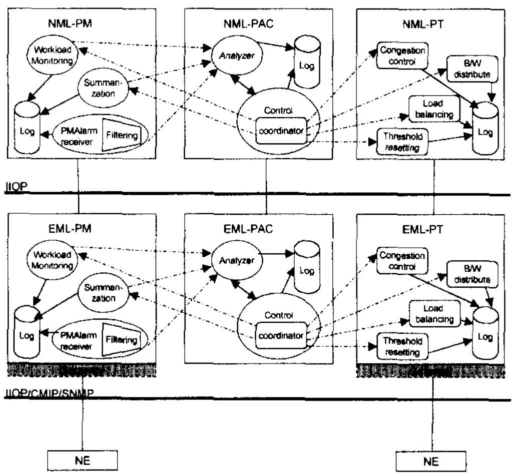


Fig. 1 Proposed Performance Monitoring Architecture

with NEs. The PT module performs the functions that improve the network performance or compensate according to the PAC results. These processes are performed according to the performance management policies[9].

The gateway in Fig.1 is necessary only for NEs with CMIP(TMN)[9] or SNMP which do not have distributed processing environment of TINA. The performance management function has tight relationships with configuration management, connection management and fault management. The proposed performance management architecture can used to enhanced network performance and guaranteed-QoS to users.

1. Performance Monitoring(PM)

Fig. 2 shows the EML-PM architecture in detail. The NML-PM architecture is similar to the EML-PM. The difference between EML-PM and NML-PM is that NML-PM monitors the resources of network(e.g. integrates measured data from each node(NE)) and EML-PM monitors the resource of a node(NE). The interface i_SupportingData is used to receive the performance measurement points and their information from NTCM(Network Topology Configuration Management) in configuration management. The EML-PM receives the values of performance management alarm(such as threshold-cross alert) from the notification server of the fault management module through the interface *i_PMAlarmR*. The interface *i_PMAlarmR* is not used in the NE with SNMP. In other words it is used only TMN equipments. The interface *i_MonitoringReq* is used to send requests of the performance monitoring to the specified measurement point. If the measurement point is TINA-based NE, then the *i_MonitoringReq* is connected directly to the NE. If the measurement point is non-TINA NE, however, then the *i_MonitoringReq* is connected to the NE through the gateway. The interface *i_MoniteringRes* is used to receive the responses of the requests through *i_MonitoringReq*. The interface *i_SendMValues* is used to deliver the monitored values to the PAC module. The interface *i_EMLPMDMonitoring* is used to receive the request of performance monitoring from the NML-PM. The Monitor in EML-PM determines the creation of either the workload monitoring function thread or the summarization function thread according to the *i_EMLPMDMonitoring* parameters. The filtering functionality of the PMAlarm receiver determines either PM alarm (such as threshold-crossing alert) or PT result according to the *i_PMAalrmR* parameters. The filtering functionality of workload and summarization classifies the kind of monitored values, such as cell count, QoS, OAM, etc.

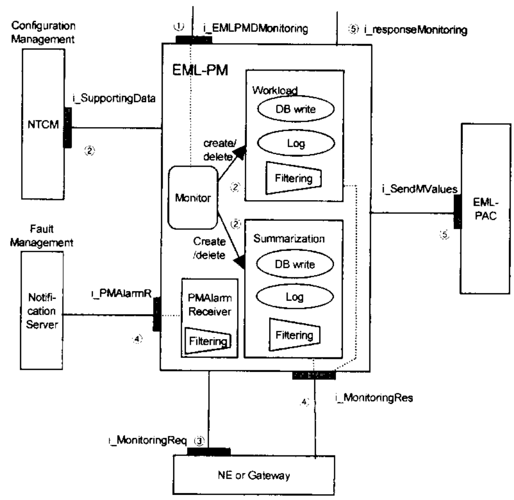


Fig. 2 Proposed Performance Monitoring Architecture

2. Performance Analysis and Control (PAC)

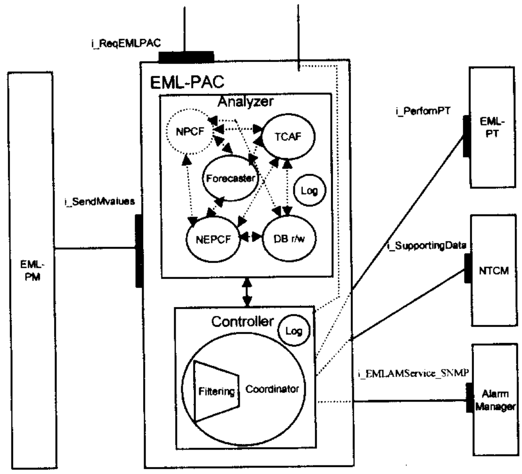
Fig. 3 depicts the EML-PAC architecture in detail. The NPCF(Network Performance Characterization Function) generates the report to characterize the end-to-end performance of the dedicated networks, which includes network interface-to-network interface, and network interface-to-inter-network interface(point of termination), in relation to long-term accuracy and availability objectives. Generally, this function is not used to in EML. The NEPCF(Network Element Performance Characterization Function) provides the processed and analyzed performance data from the measurements of the current cell count, and the history counts and the threshold-cross alert to support the evaluation of the performance level of the entity. The TCAF(Traffic Capacity Analysis Function) generates the reports of the estimated level · of offered traffic that can be carried by the current resources at the desired level of QoS.

If it is occur that the network or NE performance alarm such as threshold-across, over-utilization, etc, then it is notify to the EML-AM through the interface *i_EMLAMService_SNMP*. The interface *i_ReqEMLPAC* is called by the NML-PAC or operator which analyzes the performance of sub-network or NE(mainly related to the long-term analysis such as traffic amount of the whole network). If the network topology needs reconfiguration or expansion, then the interface *i_SupportingData* is called. The network topology reconfiguration or expansion is related only to the long-term analysis. To cope with the results of analysis, the coordinator of controller requests the suitable operation(such as the congestion control and the load balancing in the PT) to other components(PM, PT or upper layer performance components). The architecture of NML-PAC is similar to the EML-PAC.

To analyze the performance of network and network elements, we consider the following:

- link/port throughput and utilization(rate)
- node(NE) throughput and utilization(rate)
- network throughput and utilization
- connection rejection ratio
- transmission error rate
- the probability of the rejection ratio, the probability of the threshold-cross(over utilization)

To compute the link/port throughput and utilization, as a first step, we measure amount of traffic(cell counting: number of incoming and outgoing cells) on each link(port) through the PM. As a second step, the link/port throughput and utilization are computed by the number of measured cells over time and by its throughput over total ones capacity(maximum speed) each other in the PAC. The node throughput and



NPCF : Network Performance Characterization Function
NEPCF : Network Element Performance Characterization Function
TCAF : Traffic Capacity Analysis Function

Fig. 3 Proposed Performance Analysis and Control Architecture

utilization(rate) are computed by the sum of each link/port throughput and utilization(rate) in a node. And the network throughput and utilization are computed by the sum of each node throughput and utilization in network.

The connection rejection ratio and transmission error rate are given by the simple formulas below:

$$connection\ rejection\ ratio = \frac{\sum rejected\ connections}{\sum accepted\ connections + \sum rejected\ connections}$$

$$transmisstion\ error\ rate = \frac{\sum rejected\ PDUs}{\sum sent\ PDUs}$$

The probability of the rejection ratio and the probability of the threshold-cross(over utilization) are given by the formulas as shown below:

$$P_r[\,rejection\ ratio > Treshold_{rej}] = \frac{number\ of\ the\ condition[\,rejection\ ratio > Threshold_{rej}]}{number\ of\ total\ measurements}$$

$$P_r[\,utilization\ rate > Threshold_{util}] = \frac{number\ of\ the\ condition[\,utilization\ rate > Threshold_{util}]}{number\ of\ total\ measurements}$$

From the above performance analysis, we can apply these results of analysis for executing PT or network expansion and reconfiguration. And PAC reports the performance alarm to the fault management if the performance of network or NE degrade.

3. Performance Tuning(PT)

The goal of PT executes exact actions for performance correction. Fig. 4 describes the EML-PT architecture in detail. The interface *i_ReqEMLPT* is called by the NML-PT. The tuner of EML-PT creates the appropriate thread that is specified by the parameters of *i_ReqEMLPT* or *i_PerformPT*. The interface *i_SupportingData* updates the NTCM information by the results of the PT. The interface i_Tuning is used to send the requests of the tuning mechanism to NE or gateway. As the functions of PT, we consider threshold resetting, bandwidth distribution, load balancing, congestion control, etc. Currently, the performance tuning functions are not completely implemented.

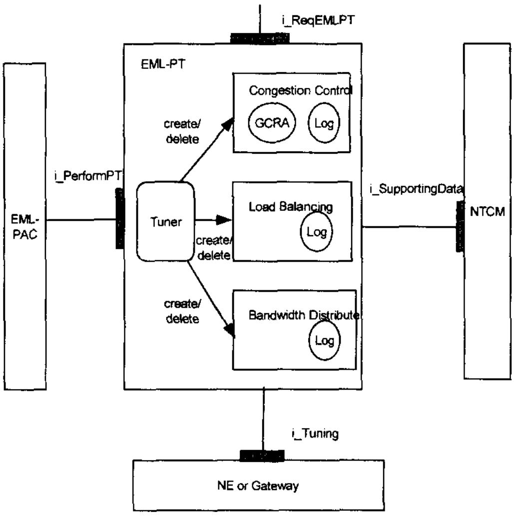


Fig. 4 Proposed Performance Tuning Architecture

IV. Interoperation with Configuration Management and Connection Management

In order to manage the performance of the network, performance management functions have to obtain the information of measurement points. Therefore performance management functions need to interoperate with the NTCM of the configuration management[9].

Information of measurement points obtained from the NTCM are information of termination points such as termination point name(DN or OID), assigned B/W, traffic parameter, QoS parameter, etc. Information of connections existed in the CP of the connection management.

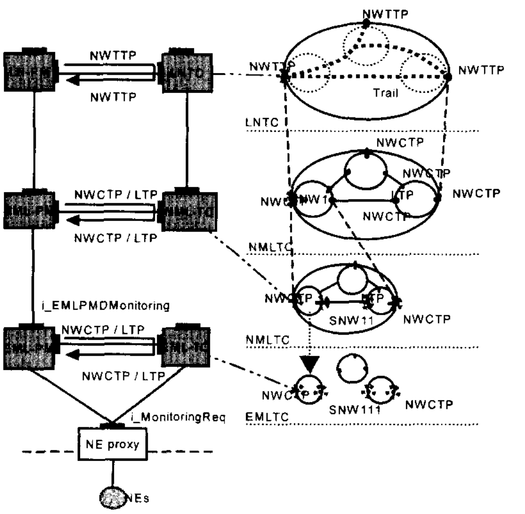


Fig. 5 Interoperation between PM and NTCM to obtain termination point

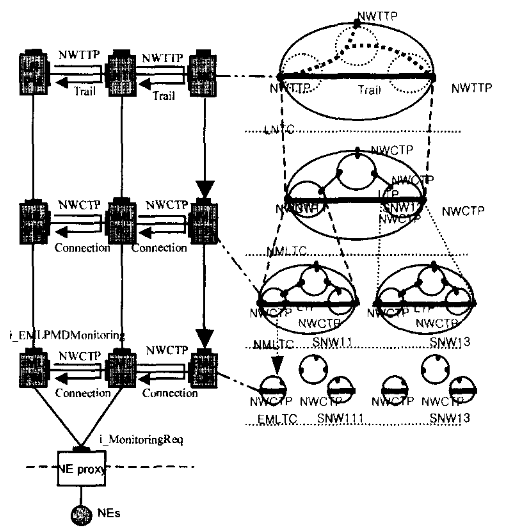


Fig. 6 Interoperation among PM, NTCM and CP to obtain termination point

Therefore information of connections requests to the CP. But performance management functions cant directly access the CP because the TINA recommended that the exchange of information between management domains perform through the NTCM.

Fig. 5 and fig. 6 illustrate interoperation among the performance management function, the NTCM and the CP to obtain the information of the termination point and the information of the connection each other.

V. Applicability Example of the Implemented Performance Management Functionality

1. Performance Management Scenario

Fig. 7 depicts overall performance management scenario. Fig. 7(a) shows PM(performance monitoring) scenario. Fig. 7(b) shows PAC (performance analysis and control) and PT (performance tuning) scenario.

The PM scenario, in Fig. 7(a), depicts procedure of the traffic monitoring for the performance assessment of NE and network in general. The PAC and PT scenario, in Fig. 7(b), depicts procedure of the analysis and tuning for the assessment of performance and activity of performance assurance of network and NE.

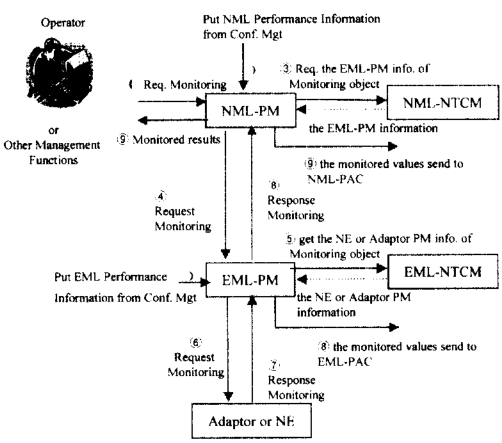


Fig. 7(a) PM scenario

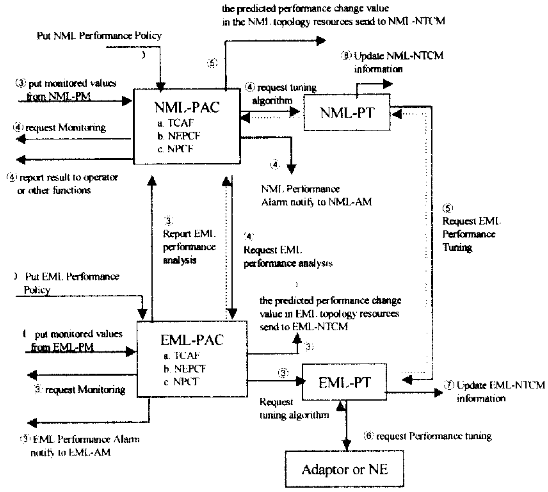


Fig. 7(b) PAC and PT scenario

Fig. 7 Performance Management Scenario

2. ATM Network Monitoring

We applied the implemented performance management functionality to real network with SNMP and TMN. The test environment consists of NML-PM, EML-PM, and NE with SNMP or TMN agent. The case NE with SNMP, the EML-PM includes TINA-SNMP. The case NE with TMN, the EML-PM includes TINA-TMN gateway[10].

The NML-PM and the EML-PM are implemented with single process and multithread architecture, respectively. When the performance management components installed by the configuration management, the Monitor thread and PMAlarmReceiver thread of NML-PM/EML-PM are drived. Where the PMAlarmReceiver thread is created only NE with TMN agent. At received the monitoring request from NML-PM or operator, the Monitor thread of the EML-PM creates workload thread or summarization thread that operates the performance monitoring. The NML-PM and the EML-PM are running in the same system.

As the real network element(managed system), we used the Fore ASX-200 ATM switch with SNMP agent. The performance monitoring function performed the cell counting for any

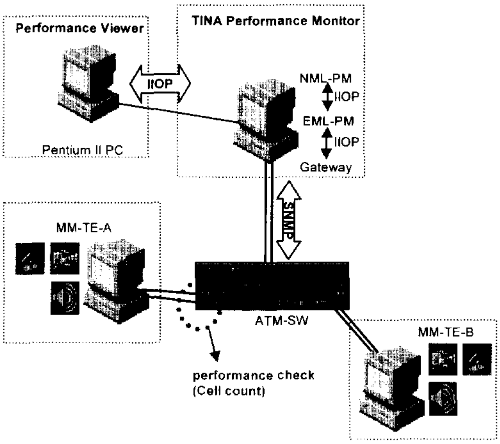


Fig. 8 Network configuration for the evaluation of Performance Monitoring

incoming port and any outgoing port. To display the monitored cell count graphically, we implemented a simple traffic monitoring viewer. Fig. 8 shows the test environment.

The proposed architecture is implemented and installed in the Sun Ultra 1 Workstation with Solaris 2.5.1. The performance viewer is installed in the Pentium II PC with NT 4.0. Fig. 9 shows the overall test flow. The viewer gets the initial configuration information through the NTCM when the network configuration is set up for the performance monitoring.

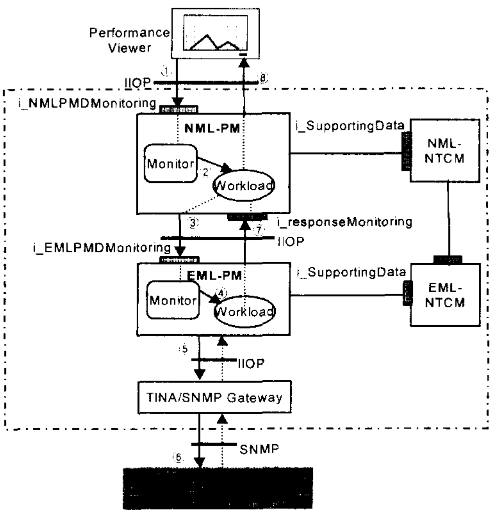


Fig. 9 Performance Monitoring Test Flow

The test sequence is as follows:

① The *i_NMLPMDMonitoring* of NML-PM is called by the performance viewer.

② The Monitor of NML-PM creates the workload monitoring thread.

③ The created workload thread of the NML-PM calls *i_EMLPMDMonitoring* of the EML-PM.

④ The Monitor of EML-PM creates the workload monitoring thread.

⑤ The created workload thread of the EML-PM calls *i_MonitoringReq* of TINA-SNMP gateway. The TINA-SNMP gateway performs abstract translation of the operations from TINA to SNMP function.

⑥ The TINA-SNMP gateway performs realistic workload monitoring operation (SNMP_Get) to NE.

⑦ The monitored values are reported to EML-PM through *i_MonitoringRes*, which are reported to the workload thread of the NML-PM through *i_responseMonitoring*.

⑧ The viewer displays graphically the monitored values.

In order to apply the implemented performance functionality to a NE with TMN agent, we used TINA-TMN gateway between the EML and NE[9]. And because we have only SNMP-NEs, we also used TMN-SNMP gateway between TINA-TMN gateway and NE[10]. The tested environment is the same as the above one and the tested results are the same as the above one[9].

With the results of measurement above, we can analyze performance of the network or network element using the formulas in the section 3.2. And by using the following rule, we can dynamically manage the capacity of a link or transit path.

```
if(threshold crossing on link1)
   if(allocated capacity+20% > some limit) then
      notify_to_NML_or_operator(·)
   else
modify_link1_capacity(link1,
allocated capacity + 20%)
```

Fig. 10(a)~(b) shows the monitored results of the incoming and outgoing cell counts on a port of the FORE ASX-200 ATM Switch. Fig. 10(a) shows a graph when the CCD cameras of TE-A and TE-B(Fig. 8) are started to capture moving objects after the performance monitoring. Fig. 10(b) shows a graph when the moving objects disappeared.

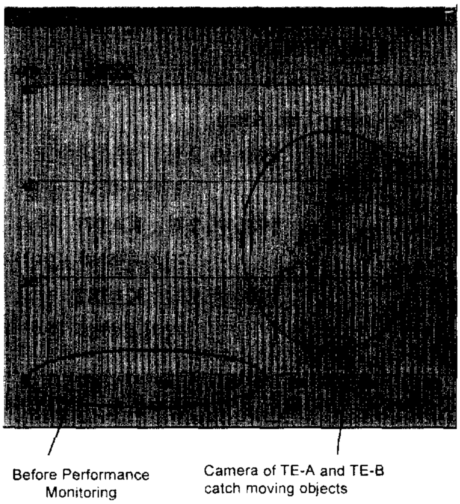


Fig. 10(a) Monitored traffic with start of movement

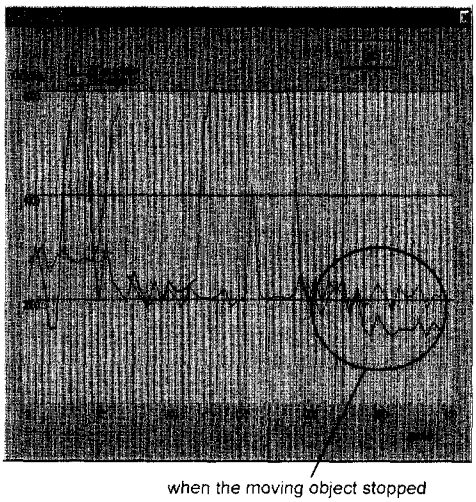


Fig. 10(b) Monitored traffic with stop of movement

Fig. 10 The result of performance monitoring (cell counting)

Ⅵ. Conclusion

We proposed a performance management architecture based on the TINA distributed processing environment and the definitions of the performance management function in TMN. We implemented the performance monitoring function according to the proposed architecture, and evaluated its operation. And we designed and implemented functional blocks of a computational component of the performance management architecture based on the multiprocess and multi-thread architecture. The performance management functions have been implemented with C++ language and Orbix 2.3c that supports CORBA 2[11-12].

We evaluated the operations of the implemented performance management functions by applying it to the performance monitoring of the Fore ASX-200 ATM switch. By this results, we confirmed that the proposed performance management architecture can be applying to heterogeneous networks(with SNMP or TMN).

As further study, the performance analysis and control (PAC) and performance tuning(PT) will be updated, and the performance monitoring will be enhanced. After these functionalities are completely implemented, we expect that the proposed performance management architecture can be applied to enhance network performance and to provide guaranteed-QoS to user.

Reference

[1] ITU-T Rec. M.3010, "Principles for a Telecommunications Management Network," Oct. 1992.

[2] Divakara K. Udupa, "*Network Management Systems Essentials*," McGraw-Hill, 1996.

[3] ITU-T Rec. M.3400, "TMN management functions," Apr. 1997.

[4] ITU-T Rec. X.739, "Information Technology-Open Systems Interconnection-Systems Management : Metric Objects and Attributes,"

Nov. 1993.
[5] ITU-T Rec. X.738, "Information Technology-Open Systems Interconnection-Systems Management: Summarization function," Nov. 1993.
[6] TINA-C Deliverable, "Management Architecture Ver. 2.0," Dec. 1994.
[7] TINA-C Stream Deliverable, "Computational Modeling Concepts Ver. 3.2," May 1996.
[8] TINA-C Deliverable, "Engineering Modeling Concepts Ver. 2.0," Dec. 1994.
[9] Seong-Woo Kim, Young-Tak Kim, "TINA-based Performance Management Architecture for Broadband Networks," *ICOIN-14*, pp 2C-1.1~2C-1.6, Jan. 2000.
[10] Ho-Cheal Kim, Seong-Woo Kim, "Hae-Joon Shin, Young-Tak Kim, Integrated Network Management with TMN & SNMP Agent and TINA Manager," *APNOMS 99*, pp. 393-404, Sep. 1999.
[11] OMG, "http://www.omg.org".
[12] IONA, "Orbix Programmers Guide," *IONA Tech.*, Oct. 1997.

김 성 우(Seong-Woo Kim) 정회원

1997년 2월 : 경일대학교 전자공학과 졸업
1999년 2월 : 영남대학교 대학원 멀티미디어통신공학과 졸업 (공학석사)
1999년 3월~현재 : 영남대학교 대학원 정보통신공학과 박사과정
<주관심 분야> TINA, SNMP, TMN, NGI, ATM/B-ISDN

김 영 탁(Young-Tak Kim) 정회원

1984년 2월 : 영남대학교 전자공학과 졸업
1986년 2월 : KAIST 전기 및 전자공학과 졸업 석사
1990년 2월 : KAIST 전기 및 전자공학과 졸업 박사
1990년 3월~1994년 8월 : 한국통신 통신망연구소 전송망구조연구실 선임연구원
1994년 9월~현재 : 영남대학교 공과대학 정보통신공학과 부교수
<주관심 분야> Broadband networking, ATM/B-ISDN, NGI, TINA, SNMP, TMN, MPLS