트리거를 이용한 네트워크관리프로그램 자동생성 기능을 가진 능동적인 네트워크 관리 시스템☆

Active Network Management System with Automatic Generation of Network Management Program using Triggers

신 문 선* 이 명 진**

MoonSun Shin MyongJin Lee

요 약

네트워크 관리는 다양한 네트워크 장비들을 적절하게 구성하고 운용하는 것이다. 대부분의 네트워크 관리 시스템은 장치구성 관리자, 성능관리자, 장애 관리자 등과 같은 기본부분으로 구성된다. 최근 광역의 개방형 인터넷의 등장으로 네트워크의 규모가 커지고 복잡해짐에 따라 다양한 네트워크 장비 및 호스트들에 대한 관리 역시 복잡해지고 중요하게 되었다. SNMP기반의 네트워크에서 다양한 종류의 네트워크 장비들을 관리하기 위한 네트워크 관리 프로그램을 개발하기 위해서는 많은 비용과 시간이 소모된다. 따라서 네트워크 환경을 확장하고 다양한 네트워크 장비들을 효율적으로 제어하기 위해서 이 논문에서는 트리거 기반의 네트워크 관리프로그램 자동생성기를 가지는 능동적인 네트워크 관리시스템을 제안한다. 트리거는 이벤트 컨디션 액션의 능동규칙으로 표현되어지며 네트워크 환경에 상태 변화가 생기는 경우 이를 감지하는 역할을 한다. 제안된 능동적인 네트워크 관리 시스템은 네트워크 엘리먼트들을 추가하거나 변경하는 경우 트리거 규칙이 활성화되고 네트워크관리시스템의 구성요소들이 상호작용하여 SNMP라이브러리에서 제공되는 정보들을 이용하여 새로운 네트워크 장비 관리 프로그램을 자동으로 생성한다. 능동적인 네트워크 관리 시스템은 자동생성된 네트워크 관리 프로그램을 이용하여 SNMP 네트워크 구성을 효율적으로 확장하는 것이 용이하다. 또한 ANMS 는 네트워크 관리프로그램의 개발기간 단축과 비용절감의 효과를 가져온다는 것을 실험을 통해서 확인 할 수 있었다.

Abstract

Network management involves configuring and operating various network elements in a suitable manner. Generally, a network management system can perform basic functionalities such as configuration management, performance management, and fault management. Due to the open structure of the Internet, the volume of network traffic and the network equipment used have increased in size and complexity. Therefore, it is expensive and time consuming to develop a network management program for heterogeneous network equipment in an SNMP based network. In order to facilitate the management of network environments and the control of heterogeneous devices in an efficient manner, we propose an Active Network Management System (ANMS) comprising an automatic generator that uses triggers to generate a network management program. The concept of triggers can be represented through event condition action rules performed in response to a change in the status of a network environment. The proposed ANMS comprises basic components for real time network management and also includes an automatic generator (AG). When the ANMS is monitoring network elements that are newly added or changed, a trigger rule is activated and these components are then able to collaborate and automatically generate a new network management program by using the information provided along with the SNMP libraries. Our method is useful for expanding the network structure and replacing network equipment. Through experiments, we have proved that our ANMS is useful when new network objects are added or changed in the network environment to expand the network structure.

* 정 회 원 : 건국대학교 컴퓨터시스템학과 강의교수
msshin@kku.ac.kr
** 정 회 원 : 가림정보기술(주) 대표이사
mjlee@galimit.com

[2008/04/14 투고 - 2008/04/29 심사- 2008/10/15 심사완료]
☆ This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-2-D00849)

Further, we have verified that our ANMS system reduces the time and cost required to develop a network management program as compared to the manual method used in existing network management systems.

☞ keyword : Active Network Management System, Active Rules, Trigger, Event Condition Action, Automatic Generator, 능동적인 네트워크관리시스템, 능동규칙, 트리거, 사건 조건 행위, 자동 생성기

1. Introduction

Nowadays, more efficient methods of managing networks and services are required because their scale and complexity is increasing. Network management has thus gained considerable importance, and it is becoming an increasingly difficult task. Initially, network management mainly involved determining the connection status of an end to end network using the Internet Control Message Protocol (ICMP). However, nowadays, information about network objects such as nodes, interfaces, and services is also desired. In addition, a standardized network management scheme for managing networks is required in order to support different network devices in a common manner. The Internet Engineering Task Force (IETF) proposed the Simple Network Management Protocol (SNMP)[1] as a standardized specification for the convenient management of the Internet. Currently, SNMP is widely used for network management. Due to the open nature of the Internet and the widespread use of the World Wide Web, most information systems are implemented in a network environment comprising various network devices. Due to the increasing scale and complexity of network architectures, the paradigm of network management has gained considerable importance. SNMP based network management has several advantages such as a simple structure, ease of implementation, and interoperability. However, it also has certain limitations in network management and operation; these limitations have become apparent with the advent of high speed telecommunication networks.

In this study, we propose an Active Network Management System (ANMS) that automatically generates a network management program in an SNMP based network environment using trigger rules. The proposed ANMS provides basic functionalities such as configuration management (CM), performance management (PM), and fault management (FM) for real time network management. In addition, it also includes an automatic generator (AG) that comprises four handlers, namely, NE Basic Info Handler, MIB Handler, Template Handler, and Operation Handler. The AG generates a network management program for new network objects that are added to an existing network environment. When the network environment is changed, a trigger rule is activated. Depending on the type of event, the ANMS checks the existing conditions and performs a corresponding action to generate a new network management program. These components work together and generate a network management program automatically using the information provided along with the network equipment and SNMP libraries. Our active network management approach is useful for improving the error rate when developing a network management program; further, it can efficiently add new network devices.

This paper is organized as follows. Section 2

discusses related works and describes their drawbacks. We introduce the information model for the concepts of triggers and active network management in Section 3. Section 4 describes the use of ANMS to automatically generate a network management program for heterogeneous network devices and interfaces to an SNMP based network management system and its architecture. Section 5 explains the comparison between our approach and existing methods through experimental results. Finally, Section 6 presents the conclusions.

2. Related Works

Recent developments in telecommunication technology, especially the advent of high speed telecommunication networks, have enabled increased network capacities; however, network structures have also become more complex. The development of network management applications becomes increasingly difficult because of the large number of network managed objects. It is necessary for a network management application to be able to manage network components automatically. It is practically impossible to develop a network management application that can consistently and automatically manage heterogeneous network managed objects.

Many application programs make use of a Web based structure to provide various types of data that is distributed over the Internet and is platform independent. Many researches are attempting to integrate various existing management protocols and tools by applying web techniques to network or system management[6][7]. However, these researches are not focused on management based on automated tools. Developers still prefer manual approaches when developing network management programs that can handle changed network managed objects. Developing such network management programs for new network elements requires a long time and is expensive. In addition, a network manager spends considerable time modifying errors. Commercial network management systems such as OpenView[10] and MRTG[11] generate such network management programs manually.

In particular, due to the disadvantages of SNMP based network management systems, many researchers have applied XML as a scheme to transfer and process the large amount of data generated in a vast network effectively[2][3]. These researches describe information using XML and transfer these XML documents via HTTP[3][4]. When a client application processes data for storage in a database, a standardized XML format is used [5].

This study focuses on how to solve these abovementioned problems that lead to an increase in the cost and time required to develop network management applications. Whenever the network configuration of a network environment changes, the network manager must modify or create a new management application to deal with the new network elements. Therefore, these tasks increase the cost and time required to develop a network management application and distribute it to the newly added network devices. We require a tool to generate a network management program automatically for newly added network elements in a large complicated network environment.

In this study, we propose an active network management system that can automatically generate a network management program for new network managed objects. Our proposed system can reduce the development cost and time required and also reduce network management errors.

3. Network Managed Object Model

In this section, we present the information model of network managed objects for the proposed ANMS. It is necessary to generate information about the target managed objects for the expanded operation of the SNMP libraries that are essential in the new network configuration approach. Therefore, we redefine managed objects and expand the MIB (Management Information Base) structure in order to design the database schema.

[Def. 1] Network Element implies the various devices that are connected to a network, such as hub, router, printer, server, switch, bridge, etc. Network management involves configuring and operating these network elements in an appropriate manner. The network management server performs network management, and the network management system manages the network system including the network elements.

[Def. 2] Network Management System is defined as a set of NE, NMServer, and NMProgram. NE implies network elements and NMServer is a system in which the network management program is loaded. NMProgram provides services to perform network management and handle the network elements.

NMS = {NE, NMServer, NMProgram}

[Def. 3] ManagedObject are the target objects to be managed in the network environment. Generally, it is represented as a set of Nodes, Interfaces, and Services

ManagedObjects = {Nodes, Interfaces, Services}

Nodes can indicate any type of network element such as router, switch, bridge, etc. The types of interfaces are logical or physical ports of a network environment for nodes and the types of services are those services provided through interfaces such as ICMP, DNS, FTP, HTTP, etc.

[Def. 4] Error is defined as a set of NError, IError, and SError.

Error = {NError, IError, SError }

NError includes node errors such as Normal, Pingfail, SNMPfail, and Datawrong. IError (interface errors) includes Normal, Linkdown, SNMPfail, and Datawrong. SErrors includes service errors and sending failure alerts.

[Def. 5] Trigger is a rule defined as an active rule in advance. Generally, it is represented as a set of Event, Condition, and Action.

Triggers = {Event, Condition, Action}

When an event occurs or the network environment changes, active rules are evaluated to trigger the AG to generate an appropriate network element handler.

The ANMS can monitor events of the network elements that are newly added or changed. When an event occurs, the condition is checked and the corresponding active rule is triggered. In fact, all active rules use the same trigger, that is, to call the automatic generator. Events mostly involve an update, delete, or insert operation. Most events are insert events to add a new network element; these are implemented by activating the AG.

Examples of trigger rules are as follows.

Rule1: CREATE TRIGGER ADD_NE

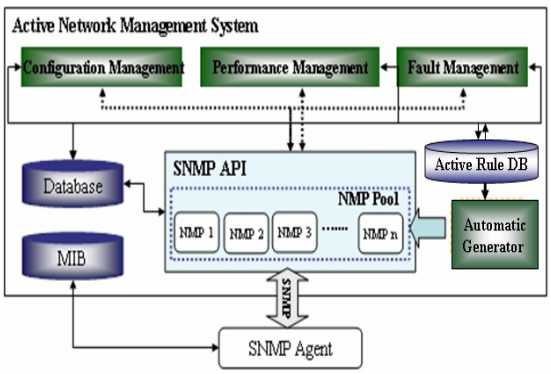
AFTER INSERT NETWORK ELEMENT

FOR EACH NETWORK ELEMENT
CALL Automatic Generator

Several rules can be predefined and stored in the active rule DB.

4. Active Network Management System

The ANMS provides an automatic tool to generate a network management program based on the active rules of triggers for newly added network objects. The network management program generated from the ANMS is used to operate and manage the network by integrating with the SNMP manager.



(Fig. 1) Components of Active Network Management System

Network managers and network program developers input managed objects that will be newly added to the network using a user interface. Then, the event condition action rules are evaluated. The AG of the ANMS generates a network management program when a trigger rule is activated. The ANMS always monitors network elements that are newly added or changed.

Figure 1 shows the proposed ANMS framework. The ANMS comprises CM, FM, PM, AG, SNMP API, and database interface for managing a vast, complicated network. Active rules are stored in the active rule database, and when a new network object is added, these rules are evaluated and the AG is triggered by them.

4.1 Configuration Management

Configuration management involves managing and configuring the network equipment. The network manager manages a group of units classified as a department or regions of departments in order to configure the network. We need to manage node configuration information such as sysDescr, sysObjectID, sysName, sysContact, sysLocation, and sysServices and interface information such as ifDescr, ifType, ifMTU, ifSpeed, ifPhyAddress, ifAdminStatus, and ifOperStatus.

4.2 Performance Management

Performance management involves optimizing the traffic performance of a network through adjustments to the network design by monitoring the network status. A network manager can perform node, interface, and service management. First, the node performance can be managed by monitoring the CPU and memory usages; if the usages cross a certain threshold, a TCA (threshold crossing alert) is generated. The TCA is a control attribute for performance management that can be used to identify the status of a network managed object. Second, we use data such as interface input usage, input throughput, input error rate, and input

discard rate for interface performance management. Finally, the service performance can be managed by using the status, node, and interface information of the current network service.

4.3 Fault Management

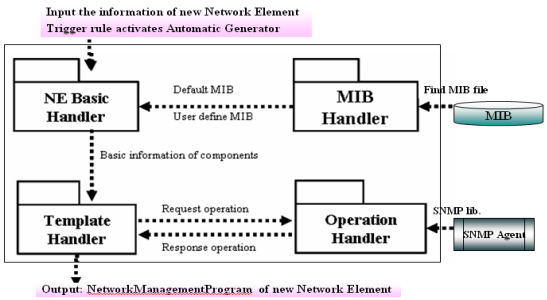
Fault management performs automatic real time monitoring of faults in a node, interface, or service in the network. Faults are classified into four types depending on their level of seriousness. Critical faults are those that cause serious problems in the system. Major faults are those that cause important problems in the system and lead to problems in the network operation. Minor faults are general faults that cause problems in the system, but do not affect the network operation. Warnings are faults that are only intimated to the network manager; these only require that the network operation be checked.

4.4 Automatic Generator

In the network environment, active rules are evaluated when events occur; the ANMS allows active rules to trigger the AG.

The AG generates a network management program automatically; it comprises the NE Basic Info Handler, MIB Handler, Template Handler, and Operation Handler. Figure 2 shows how a new network management program is generated for a newly added network element; the relationships of the four handlers of the AG are also shown.

The functions of these handlers are described in detail below.



(Fig. 2) Four Handlers of the AG

NE Basic Info Handler

This component stores and creates basic information about network management objects that will be managed by the network application. In order to create this basic information, we require the following information: (1) class name of the new network management program and program name of the network managed objects, (2) file name of the MIB used by the network manager to specify the network managed objects, (3) object name for the specific network management object, (4) acceptance or rejection of method for set operation of SNMP, and (5) acceptance or rejection of method switching over from a specific numeric data contained in the MIB to character data. We will generate a network management program based on this information.

Template Handler

The template handler supports formal information such as the template header and template tail that are commonly used in the generated network management application. The template header defines the name of the network management application and the necessary application variables. The template tail defines the

source code that configures the debugging method.

In the next section, we will analyze the performance of the framework by applying the generated network management application to an actual network management system.

MIB Handler

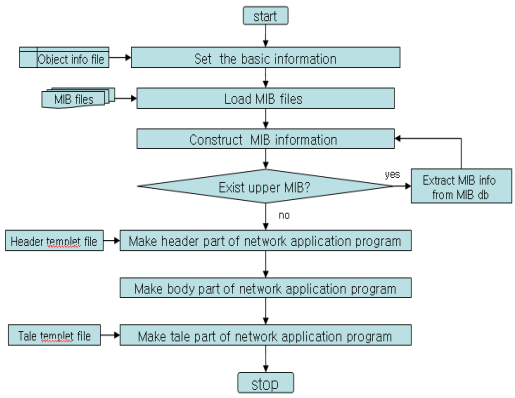
The MIB handler constructs an MIB information tree to generate a network management application. The MIB information tree represents a hierarchy of MIB objects. The MIB handler extracts the identification values of MIB objects that are a target for the network management application from the MIB information tree, and builds a tree after parsing the content of the MIB file. The MIB objects are managed and classified as single and entry objects; a single object implies that the MIB object attribute corresponds to one attribute value in the MIB information tree. The MIB information tree is created using a two step process. In the first step, the MIB handler reads the MIB files corresponding to more than one MIB filename selected in the basic data selection step (NE Basic Handler). After reading the MIB file, we generate the MIB file information tree from the read MIB file (the MIB file information tree is generated from the read MIB file). In order to generate the MIB information tree, we use default the MIB file and a user defined MIB file.

Operation Handler

The operation handler manages operations for SNMP execution. The SNMP protocol has four types of operations, namely, Get, Get Next, Set, and Trap. The Get operation reads management information such as the status and runtime of a network managed object. The Get Next operation obtains lower layer information from the hierarchical tree structure. The Set operation handles the MIB of the network managed object. The Trap operation is a threshold or event that is reported to the manager.

4.5 Process of Active Network Management

In this section, we describe the process of the proposed ANMS with AG. The flowchart shown in Figure 3 describes the overall process of the automatic generation of a network management program in the ANMS.



(Fig. 3) Process of automatic generation of network management program of new network element

First, the NE Basic Info Handler produces basic information about the network managed object. Figure 4 shows the results of the output of the NE Basic Info Handler.

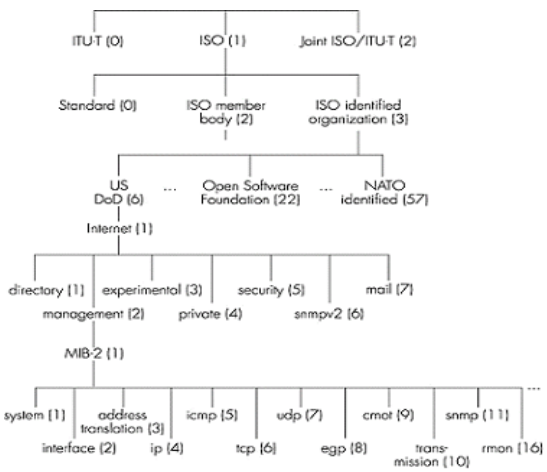
The basic information setting can be used to set the PACKAGE and java CLASS filename; it decides the names of the MIB files for the managed object. The value of OBJECTNAME

defines the managed object as a simple object or entry object, and it maps a specific value to the managed object.

The Operation handler manages operations for SNMP execution.

The MIB Handler constructs the MIB tree using the user defined MIB files and the MIB file from the MIB databases supported by network device vendors.

Figure 5 shows an example of an MIB tree. A network managed object defined by the NE Basic Info Handler must be extracted to MIB II and assigned the value of sysDescr or ifEntry to determine if it is a simple object or entry object.



(Fig. 4) Output of NE Basic Info Handler

```
PACKAGE = com.galimit.netune.snmp
CLASS = SnmpTest
LOADMIBFILE =
D:\Projects\SnmpGenerator\RFC1213  MIB.my
LOADMODULE = RFC1213  MIB
OBJECTNAME = sysDescr
OBJECTNAME = ifEntry
IGNORENAME =
SET = true
```

(Fig. 5) Example of MIB tree

***Algorithm*: Active Program Generation**
Input: network element name, default MIB file name, user MIB file name, SNMP operation
Output: Network Management Program
Method:
<Begin >
1. Load <Info File>
2. Setup Program Parameter
3. Load <Default MIB File>
4. Make Default MIB Tree
5. Load <User MIB File>
6. Add to Default MIB Tree
7. Load <Header Template File>
8. Generate Program Source {Program Header, Variables, Interface Functions, SNMP get Functions}
9. if Entry Object Exist
10. Generate Program Source {Entry Object output function}
11. Search MIB Tree for set operation enabled MIB Object
12. if set operation enabled MIB Object Exist
13. Generate Program Source {SNMP set Functions}
14. Generate Program Source {inner Class}
15. if enumeration function required
16. Generate Program Source {enumeration Functions}
17. Load <Tail Template File>
18. Generate Program Source {Program tail }
19. Save Program Source
<End>

Fig. 6 Active generation algorithm for ANMS

Finally, the Template Handler generates templates of the header, tail, and body part of the network management program called Network Element handler. NE handler is an execution file of the network handling program for the new network managed objects using SNMP libraries and MIB information.

Figure 6 shows the algorithm for the active generation of a network management program called NE handler.

```
D:\Projects\SnmpGenerator>dir
 D 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: 228C-2381

 D:\Projects\SnmpGenerator 디렉터리

2005-02-22  오후 03:55    <DIR>          .
2005-02-22  오후 03:55    <DIR>          ..
2005-02-22  오후 02:51             2,028 mib.properties
2005-02-22  오후 03:15             3,156 mibdef.txt
2005-02-22  오후 02:51    <DIR>          mibs
1996-11-06  오후 12:00           105,562 RFC1213-MIB.my
2005-02-22  오후 03:15            24,525 SnmpTest.java
               4개 파일             514,606 바이트
               3개 디렉터리  10,281,963,520 바이트 남음
```

(Fig. 7) Work directory after generating network element handler

The results of the work directory after generating the network management program is shown in Figure 7. We can see the MIB files referenced, MIBdef.txt, snmpTest.java, and snmpTest.class.

5. Experiments

We evaluated the efficiency of the generated application through the proposed ANMS. The constraint of this experiment is that it is impossible to collect information about the network objects with both the manually and automatically generated network management programs in ANMS because the network management system manages the network status in real time.

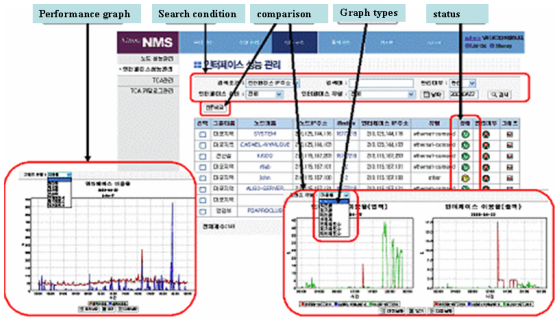


Fig. 8 User Interface of ANMS

Because of the volume of traffic, the network input constantly changes. Therefore, we compare the manually and automatically created network management programs under the assumption that the network environment and network objects are the same. We also evaluated the efficiency, development time, and error rate. In addition, we analyzed the reliability of the proposed system through experiments to check whether the system could manage the network efficiently.

Our ANMS was implemented and applied to a real network; the interface of the ANMS is shown in Figure 8.

In order to verify the network management program generated by our ANMS system and the existing method (manual method), we obtained information from the network objects, as listed in Table 1.

(Table 1) Example of network components

Node type	Node IP	Node Name
Router	211.196.xxx.127	ROUTER
Windows Server	211.196.xxx.133	KT 9Z25FJCHPIZ8

Therefore, we compared the network manageme‘nt programs generated by both the manual and automatic methods for the same network environment and network managed components.

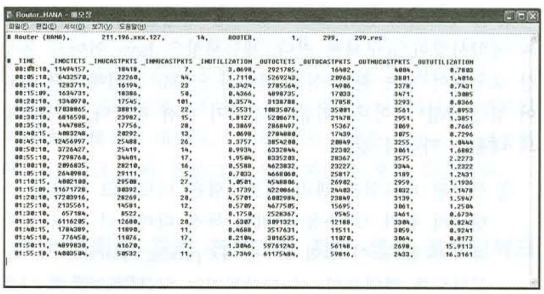
The network management system monitors the real time network status. It is impossible to obtain information about the network objects in both the manually and automatically generated network management programs simultaneously in a real network because there exist many differences in the network traffic over the course of time. The proposed ANMS can be validated by both methods as we check and compare their results.

We conducted the experiments on a computer running Windows 2000; Java was used as a programming language to develop the AG module. In addition, MySQL was used to store the network information and MIB information in a database. The following constraints related to the environment variable set for the Java class path and the SNMP library directory for the SNMP

API are used.

- Network management program generated by AG must be located in a specific directory. We set the directory as “D:\product\Radicle\” for the SNMP operations.
- The directory referenced by the MIB files and network objects information when the network management program is generated is set as “D:\product\SnmpGenerator\”.
- The SNMP library used to generate the network management program is “com.galim.net.snmp”.

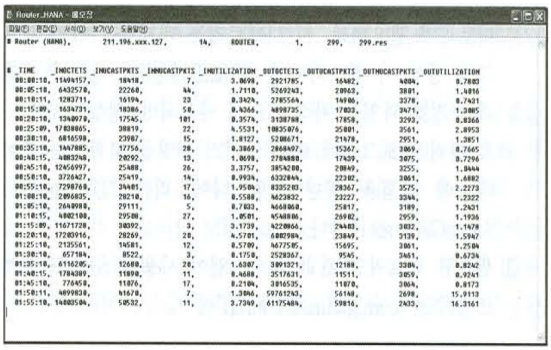
We compared the results of each network element listed in Table 1 for the validation of the AG of ANMS.



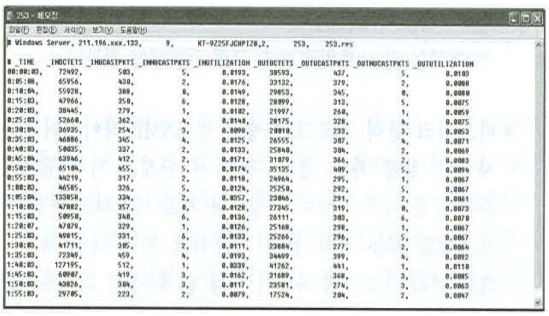
(Fig. 9) Results for manually generated network management program

We obtained the values of ifEntry objects at intervals of 5 min for 24 h. The obtained information included the number of input octacts, number of packets in input unicast, numbers of output octacts, number of packets in output unicast, and input/output ratio.

Figures 9 and 10 show the results of the network management programs generated by the existing manual process and by the AG of the ANMS, respectively. We can validate the ANMS system by comparing the results of the network object information that was produced by both methods for the same network element (router).



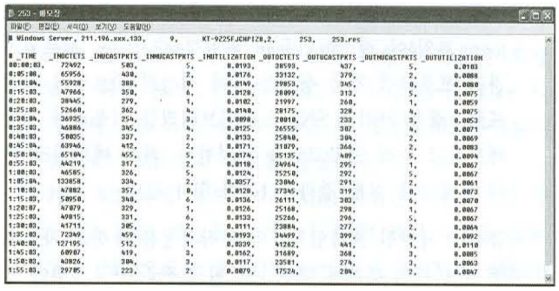
(Fig. 10) Results for automatically generate network management program



(Fig. 11) Information obtained form Windows Server for manually generation program

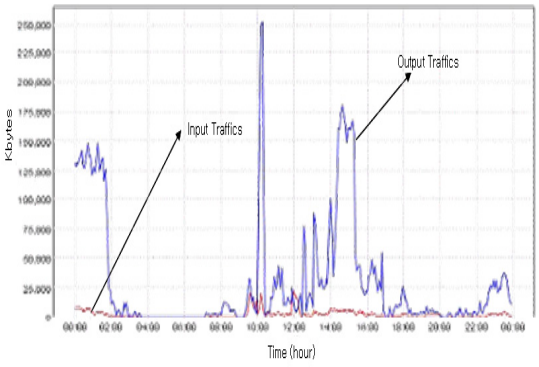
Figures 11 and 12 show the information obtained from the Windows Server, and the results were found to be the same in the case of the router. These results validate the proposed ANMS.

Although the results of both methods are identical, the proposed automatic generation of network management programs is more efficient than the existing manual method in terms of the development cost and maintenance required. We applied ANMS to a real network and found that it operated normally without any errors.



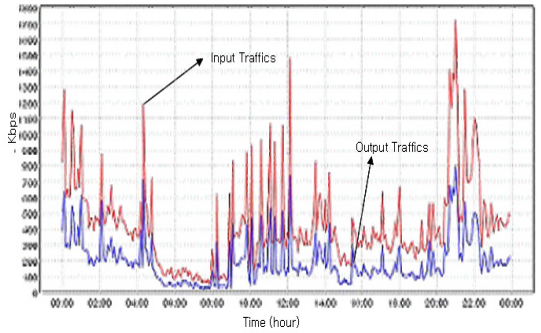
(Fig. 12) Information obtained from Windows Server for automatically generated program (ANMS)

The result obtained from the collected information for the router and Windows Server using the network management program generated using ANMS has no errors. Further, we can reduce the costs incurred for the maintenance and management of the network management system. Figure 13 shows the monitoring results of input/output traffic of the router when connected to the network using the network program generated using the ANMS.



(Fig. 13) Result of router using network program generated using ANMS

Figure 14 shows the monitoring results of input/output traffic of the Windows Server when connected to the network using the network program generated using the ANMS.



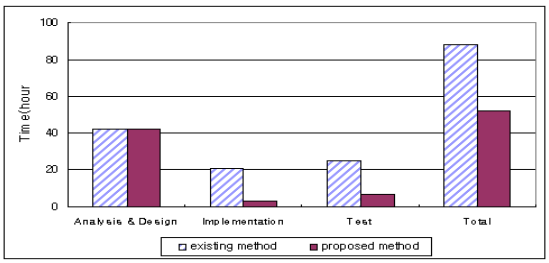
(Fig. 14) Result of Windows Server using network program generated using ANMS

We have proved that our ANMS has no errors during the execution of a generated network management program when applied in a real application (i.e., with other types of network components such as Linux server, switch, etc.). We can ensure that our application causes no errors and has high efficiency.

For a CICSO router object, we must analyze an MIB of 1416 lines with a total of 5705 lines. A total of 5705 lines implies that 4868 lines are for the common module and 839 lines are for only the CISCO module. Here, we consider the CISCO module. In order to verify the result of the CISCO module, we repeat the test step followed by error modification until no errors exist.

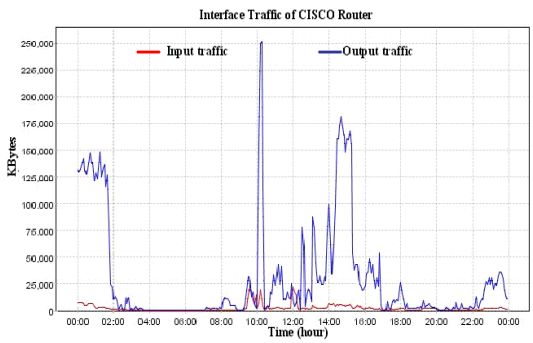
Table 2. Example of network object

Object Name	Network Object Description
CISCO Router	Cisco Internetwork Operating System Software IOS (tm) C3550 Software (C3550 I5Q3L2 M), Version 12.1(13)EA1, RELEASE SOFTWARE (fc1) Copyright (c) 1986 2003 by Cisco Systems, Inc. Compiled Tue 04 Mar 03 03:10 by yenanh



(Fig. 15) Comparison of development time between manual and active methods

Next, we employ this module in the network management system. The verifying processing terminates when this module monitors the CISCO Router and finds it functioning normally. Otherwise, we repeat this process.



(Fig. 16) Result of Input/Output traffic of CISCO router

Figure 15 shows the test results of the two methods. The existing method requires approximately 1 week for analyzing all the lines and 3 days for developing only the CISCO module. The comparison reveals that we can reduce the total time required to generate the network management program by 36% using our system.

We find that the information obtained from the CISCO router object using the network management program generating using our system has no errors. Further, we can reduce the costs incurred for the maintenance and management of the network management system. These results confirmed the usefulness of our system. Figure 16 shows the monitoring results of the input/output traffic of the CISCO router connected to the network using the network program generated using our system. Here, the upper and lower lines indicate the output and input traffic, respectively.

6. Conclusion

A network management system monitors network elements such as nodes, interfaces, and services. It allows users to access Web services conveniently by detecting faults. When most components in a network are managed manually, significant time and cost is required to develop a network management program. Automation techniques facilitate the management of a network environment and the control of heterogeneous devices. In this study, we propose an Active Network Management System (ANMS) with an automatic generator for generating a network management program using triggers. Active rules represented as event condition action rules are evaluated in response to a change in the status of the network environment; they trigger the Automatic Generator to create a new handling program for the network element.

We also presented the information model for our ANMS.

We have verified the performance of our system with respect to the automatic generation of a network management program in a real network management system.

Our ANMS reduces the cost and time required

to develop a network management program for each network equipment; further, it decreases the error rate and cost of maintenance of the network management program.

References

[1] Stallings, W.: SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. 3rd edn, Addison Wesley, Reading, MA, USA (1999)

[2] Ju, H.T., Han, S.H., Oh, Y.J., Yoon, J.H., Lee, H.J., Hong, J.W.: An Embedded Web Server Architecture for XML Based Network Management. The IEEE/IFIP Network Operations and Management Symposium, Florence, Italy (2002) 5 18

[3] Kim, Y.D., Cho, K.Y., Heo, J.H., Cheon, J.K., Cho, S.H.: Network Management System by using Transfer SNMP. Proc. of KNOM Conference, Taejeonn, South Korea, May (2001) 102 106

[4] Barillaud, F., Deri, L., Fedirum, M.: Network Management Using Internet Technologies. Proc. IEEE/IFIP International Symp. On Integrated Network Management, San Diego, CA, USA (1997)

[5] Deri, L.: HTTP Based SNMP and CMIP Network Management. Internet Draft, IBM Zurich Research Laboratory (1996)

[6] Pell, H.A., Mellquist, P. E.: Web Based System and Network Management. Internet Draft, Hewlett Packard (1996)

[7] WBEM: http://wbem.freerange.com

[8] Perkins, D., McGinnis, E.: Understanding SNMP MIBs, Prentice Hall (1997)

[9] Case, J. (et al): Management Information Base for Version 2 of the Simple Network Management Protocol. IETF, RFC 1907 (1996)

[10] OpenView: http://www.openview.com

[11] MRTG: http://people.ee.ethz.ch/~oetiker/webtools/mrtg

[12] Lee. M. J.: A Network Management System Based on Active Program Generation. Ph.D. Thesis, Chungbuk National University, Korea (2005)

◐ 저 자 소 개 ◑

신 문 선

1987년 충북대학교 전산통계학과 졸업(학사)
1997년 충북대학교 대학원 전자계산학과 졸업(석사)
2004년 충북대학교 대학원 전자계산학과 졸업(박사)
2005～현재 건국대학교 컴퓨터시스템학과 강의교수
관심분야 : 데이터베이스, 데이터마이닝, 센서네트워크, RFID보안 etc.
E-mail : msshin@kku.ac.kr

이 명 진

1984년 충북대학교 계산통계학과 졸업(학사)
1986년 숭실대학교 대학원 전자계산학과 졸업(석사)
2005년 충북대학교 대학원 전자계산학과 졸업(박사)
1989년-2001년 케이티인포텍 망관리사업팀 망관리사업부 근무
20002~ 현재 가림정보기술(주) 대표이사
관심분야 :NMS, GIS, USN , U-health etc.
E-mail : mjlee@galimit.com