A Web-based Management System for Network Monitoring

Jae-Won Choi and Kwang-Hui Lee
Dept. of Computer Engineering, Changwon National University
{jwchoi, khlee}@sarim.changwon.ac.kr

Corresponding Address:

Prof. Kwang-Hui Lee
Department of Computer Engineering,
Changwon National University,
9 Sarimdong, Changwon, Gyeongnam,
641-773, South Korea

E-mail: khlee@sarim.changwon.ac.kr
Tel: +82 55 279 7424
Fax: +82 55 279 7420
A Web-based Management System for Network Monitoring

Jae-Won Choi and Kwang-Hui Lee
Dept. of Computer Engineering, Changwon National University
{jwchoi, khlee}@sarim.changwon.ac.kr

Abstract

In this paper, we have designed and implemented a network monitoring and analysis system with efficient and integrated methods. As implemented using Java language with JDBC (Java Database Connectivity), the proposed system can be operated platform-independently and can produce various kinds of statistical information through single Web interface. A manager is able to utilize the functions such as acquisition of system information, reports on the condition of network configuration, real-time monitoring of backbone traffic, fault management, observing of connection state to the servers, measurement of system usage and so forth. Because of providing various statistical reports, this system is expected to present the basis of the extension of a legacy network and the direction of redesigning network configuration.

1. Introduction

Today, the number of computer and Internet users is rapidly increasing by advancement of information technology. E-mail and information retrieval were the most of Internet use in the past, but recently entertainment traffic such as file transmission of bulk, game, movie, and on-line TV is increasing[1]. So users require application services that have the faster speed and better quality. These imply quantitative and qualitative growth and raise the necessity that network should be managed efficiently and its performance should be improved. Therefore, the construction and operation of management system is necessary to prevent schemeless use of resources and present the basic data for expansion or investment[2].

Network management systems that have been studied and developed up to now tend to incline to the analysis of equipment itself or traffic. They have some drawbacks. They are difficult to use and do not furnish integrated environment. Also, they have several limitations that are operated on specific operating system and software[3-6].

In this paper, we have designed and implemented a management system to satisfy above necessity and complement drawbacks of existing researches. We intend to present the methods to monitor and manage the network and system resources more efficiently by consistent and integrated means.

2. The architecture of proposed system

The proposed system is a Web-based network monitoring and analysis tool, which is easy to use. The purpose of this paper is to provide a guidepost for the efficient use of network resources and well-founded investment.

2.1 Characteristics of the system

While other previous studies have focused on particular functions of network management, the characteristics of the proposed system are listed below.
- Platform-independence with Java language[7]
- Database-independence with JDBC[8]
- Easiness to use with single Web interface
- Statistical analysis and history management with DB
- Real-time monitoring
- Fault management
- Management of main server system

2.2 Architecture of the system

In this paper, we have designed and implemented a system to access the NMS (Network Management System) through single Web interface. The architecture of the system is illustrated in Figure 1 and each function of modules is described below.

2.2.1 Network manager
A network manager can interact with this system through only Web browser and can collect information about networks and server systems wherever Internet is available. For this, a manager utilizes HTML (Hypertext Markup Language) files, various kinds of reports produced by server processes, JPEG (Joint Photographic Experts Group) image files, Java applets and so on.

2.2.2 NMS server

NMS server is the essential station, in which the manager monitors and analyzes the network. The overall system consists of Web server, DBMS, server processes, and SNMP (Simple Network Management Protocol) manager[3, 9]. Web server replies to the requests from the Web browser and transfers HTML files, image files, Java applets and so on. Because all of data for management are stored in a database and accessed with JDBC, the DBMS can be within NMS server or can be constructed and operated as separate server only for the database. Server processes are divided into two parts: daemon processes and functional modules. The daemon processes wait for connection, reply to the requests and control the execution of modules periodically. Functional modules are independent each other. They behave depending on the control of daemon processes and execute their own particular functions. The SNMP manager called by server processes requests the MIB (Management Information Base)[10] information on network devices to the SNMP agent in the corresponding equipment and returns the values to the user.

3. Design and implementation of modules

This section describes the design and implementation details of server processes and presents our experience.

3.1 Initialization module

This module provides an integrated environment for driving system. It determines the scope of administrative target, searches the equipment supporting SNMP and collects the basic information for management. A manager can decide some policies and set several items. Figure 3 shows the progress process of each step. The detail functions of each component are described as follows.

- **JDBC setting**
  In this step, a manager sets up JDBC driver, port number, accounting, corresponding E-mail address and tests connectivity depending on the kind of DBMS. After these values are saved as files, all modules composing this system can access to the DBMS using these environment files.

- **Database initialization**
  In this step, the system creates the database tables for environment, policies, the basic information of collected equipment and some other values.
  - **SERVICE** table for the equipment supporting SNMP
  - **ADDRESS** table for physical addresses
  - **IPGROUP** table for grouping IP addresses
  - **THRESHOLDS** table for the values of threshold
  - **FAULTHIS** table for the history of state changes

- **Administrative target input**
  Through this module, the user can input the individual IP addresses or the range of IP addresses for administration. This information can be provided in letter or file format.

- **Searching the equipment supporting SNMP**
  As the result of SNMP query, the information about the equipment supporting SNMP is stored in **SERVICE** table. Figure 4 is the flow diagram of this procedure.
3.1.5 Removing IP redundancy

In general, because the equipment supporting SNMP can have multiple interfaces, a network device can have several IP addresses. Also, because of the use of VLAN (Virtual Local Area Network), a network device cannot be distinguished by IP address itself only. To solve this problem, this procedure compares all physical addresses of equipment and distinguishes individual equipment whether it is identical or not. IP addresses are grouped and one of these is selected as a representative IP address of the equipment. Figure 5 shows above procedure.

3.1.6 Collecting basic information

With this procedure, the system can gather various kinds of data and create the corresponding database tables to store the information about the equipment.

- **GROUPHEAD** table for group information
- **GROUPINFO** table for individual interfaces
- **RELATEIP** table for the relationship of information

3.1.7 Selecting groups and interfaces

To keep away consumption of resources and reduce the load of system, a manager can select the equipment and interfaces with necessity to be actually managed. Finally, *Rcode* table is created, where *code* part is the number indicating a group.

3.2 Data collection module

This module collects the detail information about the equipment based on the established information in initialization step and stores them in database automatically. The process refers to **GROUPHEAD** table, **GROUPINFO** table and **RELATEIP** table to get target IP addresses. This module acquires the MIB values such as *ifInOctets*, *ifOutOctets*, *ifInErrors*, *ifOutErrors*, *ifInDiscards*, *ifInUnknownProtos* and so on using SNMP. The item such as *ifInOctets* represents the total number of octets received on the interface, so that we compose count field and compare this value with *sysUpTime* value of MIB. Considering system load and accurate analysis, this process is executed every 5 minutes in our environment and the collected information is stored in *Rcode* table with time stamp. Of course, time interval can be regulated relying on environment or requirement.

3.3 Network composition module

This module is a tool that automatically draws the network topology. With this user-friendly tool, we can recognize connection structure of network nodes and abnormal links.

3.3.1 Collecting Routing information

Routing information is gathered from *ipRouteTable* including entries such as *ipAdEntAddr*, *ipAdEntIfIndex*, *ipAdEntNetMask* and so on in MIB. This process discriminates networks, acquires connection states, and automatically forms network topology.

3.3.2 Making out layout

This process makes a administrator manage the information about network composition with gathered routing information in GUI (Graphical User Interface) environment. Layouts for a backbone or several domains can be created. And it is very easy to remove, append, and modify nodes. Layout information is saved as files that contain the type of equipment, IP addresses, interface numbers, description, connection information, and coordinates to display nodes. These files are used by present condition of backbone traffic module and fault examination module.

3.4 Present condition of backbone traffic module

This module shows the real-time traffic state of backbone through applet within a manager’s Web browser. The applet creates a thread in each link, communicates with server processes, and illustrates real-time traffic information on each link. If the excessive traffic flow happens on a link, the applet informs this fact as beep sound, text, and color change. Figure 6 shows the screen of real-time backbone traffic monitoring, so we can intuitually observe whole backbone.
3.5 Fault examination module

Through a Web browser, the state (up/down) of network equipment is monitored and the fault information is stored in \textit{FAULTTHIS} table. If there is an error, this is expressed on applet in a Web browser and immediately notified to a manager through E-mail. Like present condition of backbone traffic module, this module builds up network topology and illustrates network topology with appropriate pictures and links using the files saved by network composition module. All nodes and links cannot be represented in an applet, so we hierarchically designed this procedure. Figure 7 shows a screen shot of the running applet in a Web browser.

3.6 Threshold check module

A manager can determine warning level and danger level for normal data and error data about maximum bandwidth of each interface. The \textit{THRESHOLDS} table contains equipment code, interface number, IP address, maximum speed, type, description, as well as critical values set up by a manager. This process periodically measures the quantity of data flow and checks whether it exceeds the threshold or not. If it happens, E-mail containing this information is immediately sent to a manager and the process displays detailed information with beep sound and pop-up window. And this history is stored in \textit{FAULTTHIS} table. Utilizing the SMS (Short Message Service) of mobile telecommunication, a manager may deal with this problem more quickly and actively. At any time, a manager can inspect the history of fault or error as well. Figure 8 shows the execution of this module.

3.7 Traffic Report Module

This module is similar to MRTG (Multi Router Traffic Grapher)\cite{5} in some aspects, but the main difference between this module and MRTG is the application of critical threshold. Our graphs provide daily, weekly, monthly, and yearly statistics of network traffic loads via the maximum bandwidth of interface and determined critical values. Each level is expressed by different color and Figure 9 is the view of this report.

3.8 Traffic Analysis Module

Although this module is similar to traffic report module, this module illustrates the graph that is for a manager to inspect the traffic information within a specific term of time. Now, this module is upgraded in our NetDAS (Network Design and Analysis System) project\cite{11}.

3.9 Server connection state module

Connection information can be got from \textit{tcpConnState} which is one of the MIB entries preserving the state of TCP (Transmission Control Protocol) connection. We can utilize \textit{established(5)} state to represent real-time connection state. This applet is shown in Figure 10 as the result of analysis about protocols based on TCP.
3.10 System usage module

This module illustrates the usage of system resources in a report form. Special process existing in remote server collects the system information and transfers the data to NMS server. This process utilizes the external commands of system such as ping, nslookup, uname, dmesg, iostat, vmstat, df and so on. Figure 11 shows a Web page presenting system information and usage.

3.11 Real-time equipment monitoring module

This module is the process that shows the activation state of all interfaces and the real-time traffic load of each interface at once. Figure 12 shows dynamic graphs.

3.12 Equipment information module

This is a separate program that achieves the role of a simple MIB browser.

4. Conclusion and future work

As discussed above, we have designed and implemented the management system that monitors and analyzes network and system through single Web interface.

To implement the proposed system, we have used Java language, which is the most suitable for networking and threading and matches well with web interface. Also, the network manager can operate regardless of platform type. Since the collected information and the results of analysis are stored in database, various kinds of statistical and historic reports can be provided wherever Internet is available. We have applied administrative policy and manager’s trends to our system as well. This system is expected to present the basis of the extension of a legacy network and the direction of redesigning network configuration.

Now, performance optimization of this system remains a little. Another interesting and challenging future work is the development of a network simulation tool. Several works are in progress in our NetDAS project.

References