Improvement of Network Traffic Monitoring Performance by Extending SNMP Function

Chun-Kyun Youn
Information technology division of Honam University, Kwangju, Korea
Tel : +82-62-940-5597 Fax : +82-62-940-5715 E-mail: chqyoun@honam.ac.kr

Abstract: Network management for detail analysis can cause speed decline of application in case of lack band width by traffic increase of the explosive Internet. Because a manager requests MIB value for the desired objects to an agent by management policy, and then the agent responds to the manager. Such processes are repeated, so it can cause increase of network traffic. Specially, repetitious occurrence of sending-receiving information is very inefficient for a same object when a trend analysis of traffic is performed.

In this paper, an efficient SNMP is proposed to add new PDUs into the existing SNMP in order to accept time function. Utilizing this PDU, it minimizes unnecessary sending-receiving message and collects information for trend management of network efficiently. This proposed SNMP is tested for compatibility with the existing SNMP and decreases amount of network traffic largely

Network Traffic, SNMP

1. INTRODUCTIONS AND MOTIVATION

Standards for network management protocol are SNMP (Simple Network Management Protocol) [1] of IETF based on TCP/IP and CMIP (Common Management Information Protocol)/CMIS (Common Management Information Service) [2] of ISO. SNMP is most widely used up to now [3, 4].

The normal sending-receiving method of SNMP is that a manager sends polling to an agent on UDP (User Datagram Protocol) for a targeting MIB Object, and then the agent responds to the manager. It communicates through asynchronous server/client method. A manager can request a set of MIB (Management Information Base) simultaneously and receive results from an agent by "GetBulkRequest" PDU in SNMP v2. Except this occasion, generally a single response is received for one request to an object [5-7].

Analysis types for network management are classified a real time analysis, a basis analysis and a detail analysis. The detail analysis collects cyclic history and statistical information for given period about specific objects in order to monitor the trend of a network unlike the real time analysis [8, 9]. In case of collecting information on detail analysis, a manager repeatedly polls "GetRequest" to an agent for a MIB object and receives "GetResponse" from the agent as shown in Fig.1. The received information is stored to a database and is used for analysis of network traffic trend. Such heavy traffics occurred by iterative requests and responses between manager and agent increase the traffic load of network, add the load of manager system and cause the delays of response time [10-12].

In this paper, "Trend Information (TI)" is defined as information necessary repetitiously collect for given period for the detail analysis of networks. And I propose an improved SNMP model that decreases the network traffic load during collecting TI, implement a prototype of the proposed SNMP and analyze results after testing it.

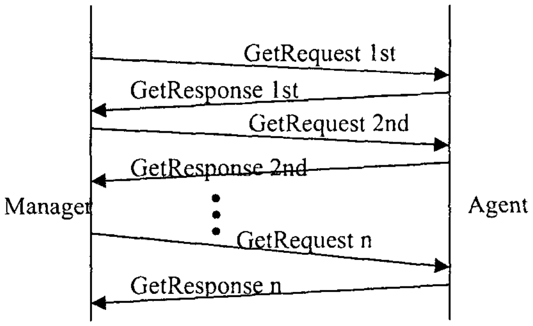


Fig.1. Sequence of SNMP for collecting trend information

The remainder of this paper is organized as follows. Section 2 briefly describes the basic concept of the existing SNMP, and section 3 explains the configuration and action principle of the proposed SNMP model. Section 4 describes the performance verification of the proposed SNMP model by executing comparison test with the existing SNMP. Finally, section 5 contains conclusion and direction for future works.

2. OVERVIEW OF SNMP OPERATION

A TCP/IP network management model consists of management system (SNMP Manager), managed device (SNMP Agent), MIB and SNMP [4-6].

The operation between Manager and Agent can be explained in terms of server and client concept. A manager sends GetRequest, SetRequest, and GetNextRequest PDUs to an agent as shown in Fig.2, and then the agent collects MIB data according to the PDU received from the manager and transmits results to the manager by GetResponse PDU. Also, agent can transmit related information to manager when some troubles or errors occurred on a managed device without request of manager [4, 5].

SNMP is a protocol used for communication between manager and agent. SNMP performs subsequent 4 functions as asynchronous request/response message protocol that operates in UDP [5, 6].

- -Get: Read MIB of agent situation and running time etc.
- -Get Next: Since MIB keeps a hierarchic structure, manager requests an agent to read the next low order MIB of relevant tree [13, 14].
- -Set: Controls devices by setting MIB values of agent.
- -Trap: Asynchronously reports to manager about important or urgent events of agent.

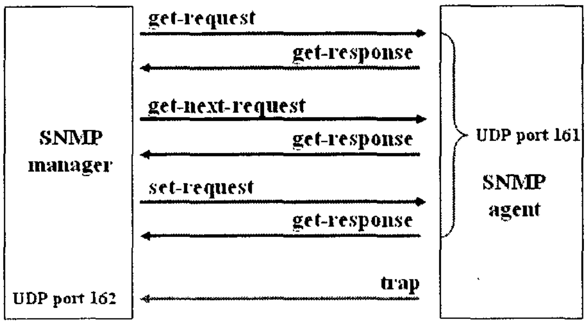


Fig.2. Operating sequence of SNMP

The composition of SNMP PDU is illustrated as Fig.3. In order to monitor TI for given period in the existing SNMP, a manager repeatedly polls "GetRequest" PDU to an agent for a MIB object like as Fig.1, and then the agent responses the relevant number of "GetResponse" PDUs. These PDU transmissions, occurred iteratively between manager and agent, increase the traffic load of network, adds significant load of manager system and delay the response of systems

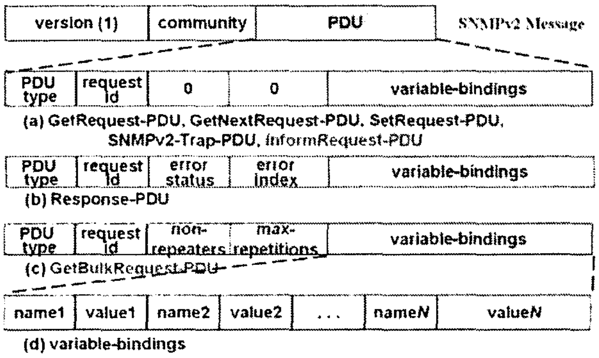


Fig.3. SNMP v2 PDU format

3. PROPOSE AN IMPROVED SNMP MODEL

3.1. Processing Sequence of the Proposed SNMP PDU

I propose an improved SNMP model as follows to solve the network traffic problems mentioned above. Fig.4 shows additional "GetTIRequest" PDU and "GetTIResponse" PDU of the proposed model. Those are added among existing SNMP PDUs (see Fig.3). So, 5 PDUs will be used on the proposed model to keep compatibility with the existing SNMP.

Fig.5 shows operating processes of the proposed model. A manager can send "GetTiRequest" PDU for measuring TI or "GetRequest" PDU except for TI to an agent by management applications. The "GetTiRequest" PDU is composed of collecting start time (start time), collecting finish time (end time), collecting cycle (time interval) fields and "GetRequest" PDU of existing SNMP.

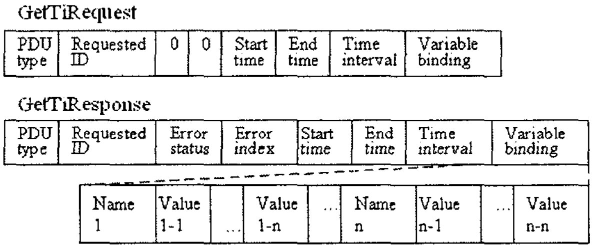


Fig.4. Operating Additional SNMP PDUs for measuring TI

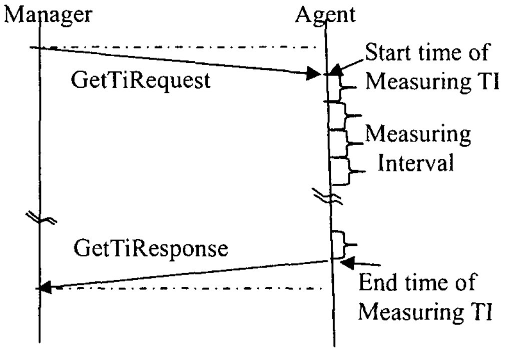


Fig.5. Operating sequence of an improved SNMP PDU for measuring TI

At first, the agent analyzes "PDU type" field of a received PDU, and then processes appropriate operations according to the PDU type. In case of the "GetTiRequest" PDU for measuring TI, an agent checks request-id, start time, end time, time interval and variable-binding fields from PDU, and collects data from the start time to the end time with the time interval. The agent creates a single "GetTiResponse" PDU containing all the collecting data, and sends it to the manager as shown in Fig.5. Otherwise, an agent receives "GetRequest" PDU and then operates as an original SNMP does.

Conclusively, the improved SNMP model sends and receives only two PDUs for certain time duration unlike existing SNMP for measuring TI. Therefore, it can decrease unnecessary network traffic and the load of manager system.

3.2. Composition and Function of the Proposed SNMP model

GetTIRequest and GetTIResponse PDU, additional PDUs for the proposed SNMP model, are composed of Start Time, End Time, Time Interval and Variable binding. The detail meaning of each item is as follows.

- -Start Time: The starting time for collecting data of related MIB variable. It is expressed by Octet string type. (Ex: '12:30')
- -End Time: The ending time for collecting data of related MIB variable. It is expressed by Octet string type.
- -Interval: The cycle for collecting data of related MIB variable. It is expressed by Integer and its unit is second.
- -MIB values of relevant variables are stored to an agent

by interval of this value (sec).

- Variable binding for GetTIRequest: Variable identifiers are indicated with name 1, name 2..., and name n, respectively. Then, a manager can request these objects to an agent at once.
- Variable binding for GetTIResponse: Express name of the objects sent by GetTIRequest identifier and measured value of the objects collected during the collecting term.

The proposed SNMP model is divided into a manager and an agent module as shown in Fig.6.

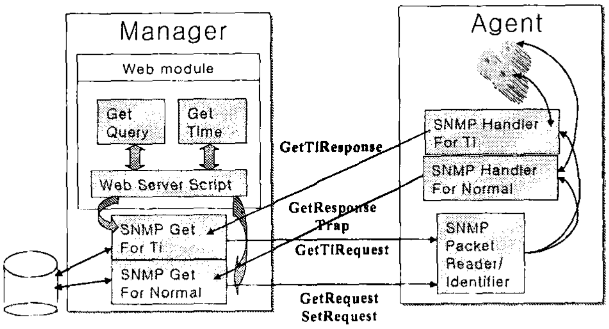


Fig.6. Proposed SNMP model for measuring TI

The manager module consists of web module and SNMP Get modules. The agent module is composed of SNMP Packet Reader/Identifier, which determines a received SNMP packet according to PDU type, and SNMP Handler, which handle it. The detail functions of proposed SNMP model are follows;

- Web module: The module offers interface to a manager through web. It has functions for setting starting time, ending time and interval. It can choose MIB object that wishes to measure [15].
- SNMP Get module: The module creates a GetRequest PDU for existing SNMP function or a GetTiRequest PDU for measuring TI according to the inputted data of web module, and sends it to an agent. It receives measured data from agent and processes it.
- SNMP Packet Reader/Identifier module: The module analyzes PDU received from manager and determines the type of PDU - GetRequest PDU or GetTiRequest PDU. Then it performs an appropriate handler.
- SNMP handler module: The module is divided into SNMP handler for TI module and SNMP handler for normal module. It collects data from resources according to information of PDU, and transmits it to the manager.

Therefore, the proposed SNMP model can keep compatibility with the existing SNMP and process measurement of TI efficiently.

4. TEST AND ANALYSIS OF RESULTS

4.1. Test Environment and Condition

In order to collect TI efficiently, the proposed SNMP is implemented. System configuration environment for testing performance comparison with existing SNMP is as following (see Fig.7).

- OS: Manager: UNIX (Solaris2.8), Agent: Linux (Redhat6.2), Measurement system: Windows XP
- SNMP version: SNMP version 2
- SNMP program: UCD SNMP v3.4
- languages for implementation: C, Shell base CGI-C compiler (GCC v3.1)
- Measuring tool: LANdecoder32, AppDancer FA™

An independent test bed is constructed on Ethernet LAN through a simple hub, which is unconnected to other network as Fig.7 in order to measure network traffics for the proposed SNMP and for the existing SNMP. A manager (IP 211. 227.240.56) and a agent (IP 211.227.240.155) communicates with SNMP, and a traffic measurement system (IP 211.227.240.35) initiates the manager system by Web program and measures network traffic by using network management tools.

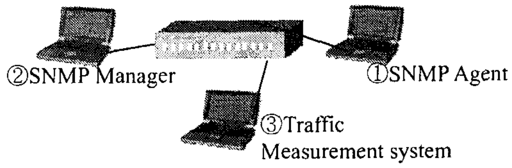


Fig.7. Configuration of Test bed

Results of collecting TI measured by the existing SNMP and the proposed SNMP are examined on this test bed Also network traffics performed by these models are compared. A MIB object ("Iso (1).Org (3).Dod (6) Internet (1).Mgmt (2).Mib-2 (1).Interfaces (2).IfTable (2).IfEntry (1).IfInOctets (10)" (total number of octets received on the interface, including framing characters)) is measured for 30 minutes every five seconds by the existing method and the proposed method. The results are illustrated on graphs.

4.2. Test Results

4.2.1 Network Traffic

1) Network traffic in the existing SNMP

The existing SNMP model acts like as Fig.1 when collect TI for the MIB object. If network traffic produced by GetRequest PDU is Tgrq and network traffic produced by GetResponse PDU is Tgrp, then traffic amount (Tf) for one sequence of measuring TI is calculated as:

$$Tf = Tgrq + Tgrp \qquad (1)$$

If an interval for collecting data is It and collecting term is Dt, then total amount of traffic (Ttf) is calculated as:

$$Ttf = Dt/It * Tf \qquad (2)$$

Expression 2 represents that the total amount of traffic for measuring TI on the existing SNMP increases as collecting term is longer and cycle is shorter.

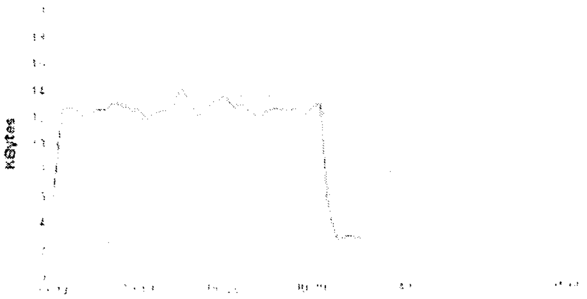
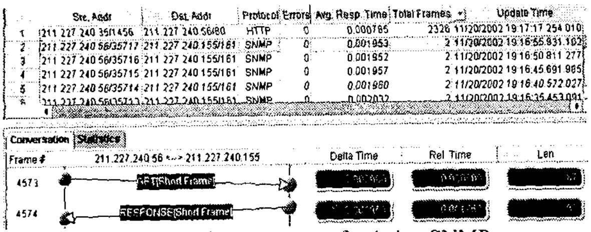


Fig.8. Network traffic measured by existing SNMP

Fig.8 represents a test result of network traffic measured for 30 minutes according to test condition in test bed by the existing SNMP. An average traffic is

12.9Kbytes/minute. However, the traffic is not consistent since there is HTTP for web program to initiate manager from the measurement system, NetBIOS for Windows XP and SNMP in the test bed.



	Src. Addr	Dst. Addr	Protocol	Errors	Avg. Resp. Time	Total Frames	Update Time
1	211.227.240.35/1456	211.227.240.56/80	HTTP	0	0.000785	2326	11/20/2002 19:17:17.254.010
2	211.227.240.56/35717	211.227.240.155/161	SNMP	0	0.001953	2	11/20/2002 19:16:55.931.102
3	211.227.240.56/35716	211.227.240.155/161	SNMP	0	0.001952	2	11/20/2002 19:16:50.811.277
4	211.227.240.56/35715	211.227.240.155/161	SNMP	0	0.001957	2	11/20/2002 19:16:45.691.985
5	211.227.240.56/35714	211.227.240.155/161	SNMP	0	0.001960	2	11/20/2002 19:16:40.572.027
6	211.227.240.56/35713	211.227.240.155/161	SNMP	0	0.002032	2	11/20/2002 19:16:35.453.091

Fig.9. Traffic Sequence of existing SNMP

Fig.9 shows traffic sequences of the existing SNMP measured by network management tool (AppDancer FA ™). The manager (IP 211.227.240.56) sends GetRequest PDU to agent (IP 211.227.240.155), and the agent responds with GetResponse PDU to the manager. SNMP frames are transmitted every five second. It shows average response time is (0+1.953msec) and frames size of two PDUs is (87 + 90B) for one sequence.

2) Network traffic in the proposed SNMP

The proposed SNMP model acts like as Fig.5 when collects TI for the MIB objects.

If network traffic for GetTiRequest PDU is Tigrq and network traffic for GetTiResponse PDU is Tigrp, then traffic amount (Tif) for one sequence of measuring TI is calculated as:

$$Tif = Tigrq + Tigrp \quad (3)$$

If an interval for collecting data is It and collecting term is Dt, then total amount of traffic (Titf) is calculated as:

$$Titf = Tigrq + Tigrp = Tif \quad (4)$$

Expression 4 represents that the total amount of traffic for the proposed SNMP does not increase even though collecting term is longer and interval is shorter.

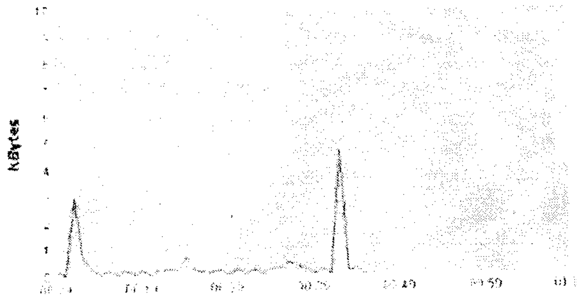
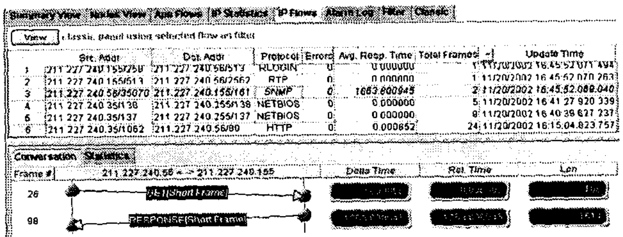


Fig.10. Network traffic measured by the proposed SNMP

Fig.10 shows that average network traffic is about 0.4Kbytes/ min in the proposed SNMP. Network traffic is relatively higher at starting and ending part. It means the pure traffic on SNMP is mainly produced only twice. The first is when manager transmits GetTIRequest PDU to agent, and the second is when agent transmits GetTIResponse PDU to manager for response. The other traffics are HTTP for web program to initiate manager from the measurement system and NetBIOS for Windows same as the existing SNMP.



	Src. Addr	Dst. Addr	Protocol	Errors	Avg. Resp. Time	Total Frames	Update Time
1	211.227.240.155/258	211.227.240.56/513	RLOGIN	0	0.000000	1	11/20/2002 16:45:52.071.494
2	211.227.240.155/513	211.227.240.56/2562	RTP	0	0.000000	1	11/20/2002 16:45:52.070.263
3	211.227.240.56/35070	211.227.240.155/161	SNMP	0	1663.600945	2	11/20/2002 16:45:52.069.040
4	211.227.240.35/138	211.227.240.255/138	NETBIOS	0	0.000000	5	11/20/2002 16:41:27.920.339
5	211.227.240.35/137	211.227.240.255/137	NETBIOS	0	0.000000	9	11/20/2002 16:40:39.627.237
6	211.227.240.35/1062	211.227.240.56/80	HTTP	0	0.000652	24	11/20/2002 16:15:04.823.757

Fig.11. Traffic Sequence of the existing SNMP

Fig. 11 shows traffic sequences of the proposed SNMP measured by the network management tool. The manager (IP 211.227.240.56) sends GetTIRequest PDU to the agent (IP 211.227.240.155), and the agent responds with GetTIResponse PDU to the manager. It shows only two PDUs were sent for 30 minutes as mentioned in fig. 5.

4.2.2 Data Consistency Test of Measured Data

I tested mutual consistency of the data collected by the proposed SNMP and the existing SNMP. For this test, two models measure the same MIB object ("IfInOctets (1.3.6.1.2.1.2.2.1.10)") to an agent for 30 minutes at every 5-second almost at the same time. Fig.12 shows the results. The data are calculated with mean value for 30 seconds using the every 5-second.

Time synchronization is necessary among manager, agent and measurement system because the existing SNMP and the proposed SNMP must read a MIB value base on synchronized time. I installed NTP (Network Time protocol) to the manager and the agent system to synchronize with standard time of "gps.bora.net" server, and the measurement system is synchronized automatically with an Internet timeserver using function of Windows XP.

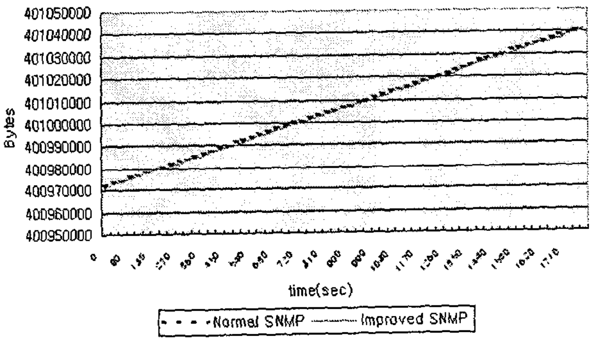


Fig.12. Data consistency test results graph

Fig.12 shows a little difference between two models since there is run-time disparity between two manager programs. Little time lag occurs here even if two web programs start almost at the same time by mouse operation. That is, this time lag induces a little difference of the measured two values. The value of the proposed SNMP, measured first, is less according to property of the MIB object. The difference is so little that two graphs are almost the same. The two graphs can be equal perfectly if the proposed SNMP graph moves a little bit to the left side. That is, the measured values of the both models will be equal if we can measure exactly at the same time.

Conclusively, the proposed SNMP satisfies data consistency, and works correctly because the values measured by two models are almost same.

4.2.2 Analysis of Test Results

In case of single transaction, the PDU sizes of two models are GetTIResponse ≥ GetResponse and GetTIRequest ≥ GetRequest according to the Figures 3 and 4. Therefore, Tigrq ≥ Tgrq, Tigrp > Tgrp, Tif ≥ Tf are led from Expressions 1 and 3. Traffic amount of the existing SNMP for single transaction is less 83bytes than the proposed model. The 83bytes can be calculated by 21bytes (108-87) and 62bytes (152-90) from the size of PDUs.

On the other hand, in case of measuring TI, which is interested in this paper, the proposed SNMP network traffic of pure SNMP is reduced about 39.3 times for 30 minutes. Total traffic of the existing SNMP is 63,720bytes (187+90) * 60 sec/5 sec*30min) according to Fig.9, while the proposed SNMP is 1,622bytes (108+1,514) according to Fig.11. Average network traffic including the other protocols is decreased about 32 times in the proposed SNMP. That is, the network traffic of the existing SNMP increases in direct proportion to the data sending-receiving frequency. However, the network traffic of the proposed model increases a little bit according to the measuring number of times, but it is not so much.

Conclusively, the network traffic of the proposed SNMP is greatly decreased compare with the existing SNMP as measuring interval gets shorter and measuring term gets longer. And the proposed SNMP keeps compatibility of function with the existing SNMP.

5. CONCLUSION

This paper is focused on measuring TI which is essential for the detail network management in SNMP network. I design an efficient SNMP that can minimize unnecessary network traffic for measuring TI, implement a prototype and test it.

The characteristics of proposed SNMP are as follows. First, it greatly decreases network traffic of measuring TI comparing with the existing SNMP, since the number of sending-receiving messages between a manager and an agent are decreased remarkably. Second, it performs extended functions for measuring TI successfully and keeps compatibility with the existing SNMP perfectly. The effect of the proposed SNMP would be great when applied to NMS that manages wide area network.

Of course my proposal has modification overhead of the orginal SNMP program, but I think it not so big weak point compare with the contribution of my proposal.

In the future, two topics will be researched. First, in case that an agent device has no hard-disk or small size of memory, how to keep the collected information for a moment in the agent and apply it to embedded systems. Secondly, whole data can be lost from agent when a GetTIResponse PDU is omitted because it is transmitted at once. So, I need to study countermeasure for occurrence of the whole data damage.

References

[1] M. Schoffstall, J. Case, M. Fedor, C. Davin: The Simple Network Management Protocol (SNMP), RFC 1157, 1990

[2] Common Management Information Protocol—CMIP, ISO 9596/ITU X.711, 1991.

[3] Amatzia Ben-Artzi, Asheem Chandna, Unni Warrier: Network Management of TCP/IP networks: presents and future, IEEE Network Magazine, Vol.4, No.4, 1990, pages 35-43.

[4] Gilbert Held: Managing TCP/IP Networks: Techniques, Tools and Security, John Wiley & Sons, 2000.

[5] William Stallings: SNMP, SNMPv2, SNMPv3, and RMON1 and 2, Vol. 3e, Pearson Addison-Wesley, 1999.

[6] Mani Subramanian: Network Management: Principles and Practice, Pearson Addison Wesley, 2000.

[7] Aiko Pras: Network Management Architectures, CTIT Ph. D-thesis series, ISSN 1381-3617; no. 95-02, 1995.

[8] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung: Design and Implementation of SNMP-based performance parameter extraction system, 1997 Asia-pacific network operations and Management Symposium, 1997.

[9] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung: A new approach to gather network management data periodically, ITC-CSCC '97, 1997.

[10] M. checkhrouhou and J, Labetoulle: An Efficient polling Layer for SNMP, proceedings of the 2000 IEEE/IFIP Network operations and Management System, 2000, pages 447-490.

[11] Min-woo Kim, Seung-kyun, Park, Young-hwan, Oh: A Study on the Polling Mechanism for Optimizing the SNMP Traffics, The journal of the Korean institute of communication sciences. 06 v.26, n.6A, 2001, pages1051-1059.

[12] Jin-young Cheon, Jin-ha Cheong, Wan-oh Yoon, Sang-bang, Choi: Adaptive Network Monitoring Strategy for SNMP-Based Network Management, The Journal of the Korean Institute of Communication Sciences.12 v.27, n.12C, 2002, pages 1265-1275.

[13] M. Rose, K. McCloghire: Concise MIB Definitions, RFC 1212, 1991.

[14] David T. Perkins, Evan McGinnis: Understanding SNMP MIBs, Prentice Hall PTR, 1996.

[15] Design and Implementation of Web Interface for Internet management System Using SNMP MIB-II, The Transactions of the Korea Information Processing Society.03 v.6, n.3, 1999, pages 699-709.