RMON 기반의 응용 서비스 트래픽 관리를 위한 방법론 연구

한정수[†] · 안성진[††] · 정진욱[†††]

요약

네트워크의 급속한 성장과 함께 사용자들이 사용하는 서비스들의 고속화와 이들에 대한 QoS 보장이 주요한 관심이 되고 있다. 이러한 이유로 네트워크의 자원을 더욱 효과적으로 관리하여 사용자들의 요구 사항을 만족시키기 위한 방편으로 네트워크 응용 서비스에 대한 관리가 필요하게 되었다. 따라서 본 논문에서는 네트워크 자원의 효율적인 사용과 더불어 사용자 관점의 응용 서비스를 더욱 효율적으로 관리 하기 위해 네트워크상의 응용 서비스 트래픽을 추출하기 위한 방법론을 제시하였고 이를 토대로 RMON MIB을 사용하여LAN 상의 응용 서비스의 트래픽을 관리하기 위한 분석 파라미터와 그 알고리즘을 도출함으로써 관리자로 하여금 손쉽게 응용 서비스의 트래픽을 추출하고 분석하는 응용 서비스 관리자 시스템을 연구하였다.

A Study on the Methodology for Application Traffic Management using RMON

Jeong-Soo Han[†] · Seong-Jin Ahn[††] · Jin-Wook Chung[†††]

ABSTRACT

With the rapid development of computer networks, how to speed up network services and to guarantee QoS (Quality-of-Service) for users has drawn much attention from computer scientists. So has the need to manage network application services in order to manage network resources more effectively and meet the users' demands. In this paper, we present methodology of determining the traffic of application services on the network, so as to manage network resources effectively and thus to make application services more user-oriented. On the basis of this methodology and by using RMON MIB we develop analysis parameters and their algorithm to manage the traffic of application services on the network, thus realizing Web-based Application Management System which allows network managers to extract and analyze the Internet application service traffic beyond the limitation of time and space.

키워드 : 응용 서비스 관리(Application Management), RMON MIB, QoS, LAN segment, 폴링 간격(polling interval)

1. Introduction

As the Internet has rapidly been developed and spread throughout the world and high-speed network environments have growingly been realized, users have tended to demand more complex and various QoS requirements [1, 2]. To meet these demands, Internet application services have turned into multimedia services which are to process large amounts of data real-time. This situation has made it desperately necessary to develop a technology of managing the traffic of application services on the network in order to use all the communication components and network resources more effectively, and maintain an organic relationship among them. All the traffics of network services including that of application services are currently managed on the Web in most cases, but a new management framework for the application traffic is being developed using the advantages of Java technology [3-5]. Along with it, MIB (Management Information Base)s for application service management are being tested [6, 7]. All these efforts to develop new network management frameworks are based on the recognition that 1) speeding-up and stabilizing networks alone is not enough to provide and support all the services efficiently and effectively, and 2) in order to meet various demands of users, new frameworks which manage Internet application services at an upper layer such as a transport layer and an application layer are needed along with the development of most stable lower layer protocols and the acceleration of network services [8, 9]. In this paper, we present a new framework to determine and mana-

[†] 준 회 원 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부
[††] 종신회원 : 성균관대학교 사범대 컴퓨터교육과 교수
[†††] 종신회원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
논문접수 : 2002년 1월 21일, 심사완료 : 2002년 4월 1일

ge the traffic of Internet application services, which can effectively complement the current network management technologies such as management structures, protocols and modeling.

Managing the Internet traffic requires us to consider some important factors as follows. First, polling frequency has to be taken into account. Network managers should choose proper polling frequency appropriate to a network situation. When the network load is heavy, for example, they have to perform polling less frequently so that an excessive management traffic might not make the network situation worse. Secondly, polling repetition interval has to be taken into account. When polling is performed twice or more, polling repetition interval has to be set considering the relationship between the preciseness of data to be analyzed and the network load. Lastly, ways to get information about managed objects should be considered [10]. Currently available or feasible are the two approaches as follows [11] :

- Approach using network layer devices
  If we use network layer probes such as RMON (Remote Monitoring Network) probe, we can listen to all data packets on the LAN, get their performance data and analyze the types of application services currently available on the Internet. To make an effective use of such network layer probes, a proper polling system should be developed [12], and appropriate polling frequency and polling repetition interval should be set to figure out and analyze the network situation and network load precisely [13].
- Approach using information provided by application layers
  We can also get performance information from log files on the server of application services. Using this method, we can analyze the log files and provide users with useful information about the services. The servers which currently use this approach include Web servers, E-mail servers, and FTP servers. This approach allows Internet application services to be managed on application layers, and thus managers are able to get precise information about users. Unfortunately, however, it does not allow managers to get a clear picture of various network situations and network load [14-16].

We are not the only one that has sought an effective and efficient way of managing the application traffic. In fact, quite a number of scientists have preceded us in this effort. Author has suggested that a certain MIB, agent for the application traffic, be used to manage a designated application service traffic [17]. [18] has indicated that it is most efficient to analyze the log file in a specific application service server. Kang have recommended that a RMON probe on a LAN segment be used to analyze various LAN performance factors for the purpose of effective management on the application service traffic [19]. Taking a step further from the research in [19], he has introduced a rule-based system designed to detect and locate an fault on the LAN [20]. What is in common between our study in this paper and the two studies mentioned above, i.e. [19] and [20], is to use a RMON probe which allows us to monitor all application service traffics on the LAN, quite unlike the methods of [17] and [18] which requires a certain application service agent MIB or log file for each application service traffic to be analyzed. What makes our study remarkable and different from the other ones of [19] and [20], however, is that we employ an algorithm designed to analyze the application service traffics in addition to a RMON probe, ensuring more effective and accurate analysis on all the application service traffics.

In this paper, we put polling frequency and polling repetition interval at the disposal of managers, and use the RMON device working on network layers to determine and analyze the application traffic on the LAN. We develop a system which determines the Internet application traffic using RMON probe and a management system which provides users with the data obtained by using the former system. In addition, we design a user interface which guarantees easy access to the data and provides various platforms of Java.

2. Analysis of Internet Application Traffic

2.1 RMON MIB Groups Used

In this paper, tree out of the nine MIB groups are used to determine the Internet application traffic. The three MIB groups are as follows :

2.1.1 Statistics Group

It is composed of objects which maintain statistics about each LAN segment observed by the RMON probe and statistics about errors. In this paper, we use this statistics group when we count the number of all the packets of the traffic currently generated on the LAN segment.

2.1.2 Filter Group

The filter group allows us to filter all the packets on a LAN segment and find out which packets match a specific one. Using this group, we are able to set the pattern of a packet on each LAN segment, and count and discover packets whose patterns match that of the packet. The filter group also allows us to store the data filtered in the process into the filter capture group. Hence we can figure out which packets have a certain pattern or a certain address and so on.

2.1.3 Capture Group

The packet capture group is where the packets filtered in the filter group are stored, and it is used to store the data of a filter observed by the RMON probe and its statistics. It usually stores the header of a packet and analyzes its data [3, 4, 16].

2.2 Analyzing Algorithm of the Internet Application Traffic

In this paper, we present two algorithms to be analyzed in order to determine and analyze the Internet application traffic : the one for application services on a LAN segment and the other for application services on a certain host in the LAN.

(1) Analysis of application services on a LAN segment

Step 1 : Identifying the interface index number

Using *ipAdEntIfIndex*, we identify the interface number I_{if} of an IP address.

Step 2 : Setting the filter group of the RMON probe as *createRequest(2)*

In order to add a new table by using an index number I_{ro1}, we set *filterStatus* and *channelStatus* of the filter group as *createRequest(2)*.

Step 3 : Setting location and value for filtering an application service

Using the index number I_{ro1}, we set the port number of an application service to be filtered at *filterPktData* of the filter group, and among the Ethernet frames we set *filterPktDataOffset* which chooses the location to start filtering. We also set the values of OxFF and OxOO at *filterPktDataMask* and *filterPktDataNotMask* respectively, which show the range of data which are set to match.

Step 4 : Choosing an interface to be monitored

Using the index number I_{if}, we set *channelIfIndex*.

Step 5 : Setting a channel group of the RMON probe

Using the index number I_{ro1}, we set *channelAcceptType* of the filter group at *acceptMatched(1)*, and *channelData Control* at on(1).

Step 6 : Setting the filter group of the RMON probe at *valid(1)*

We set *filterStatus* and *channelStatus* of the index number I_{ro1} at *valid(1)* for activation.

Step 7 : Setting the capture group of the RMON probe at *createRequest(2)*

In order to add a new table by using an index number I_{ro2}, we set *bufferControlStatus* of the capture group at *createRequest(2)*.

Step 8 : Setting the target for capturing

We set the index number I_{ro1} of the filter group which was set above at *bufferControlChannelIndex* of the capture group.

Step 9 : Setting the capture group of the RMON probe

Using the index number I_{ro1}, we set *bufferControlFullStatus* of the capture group as *lockWhenFull(1)*, *bufferControlCaptureSliceSize* as 38, *bufferControlDownloadSliceSize* as 34, and *bufferControlDownloadOffset* as 4.

Step 10 : Setting the capture group of the RMON probe as *valid(1)*

We set *bufferControlStatus* of the index number I_{ro2} as *valid(1)* for activation.

Step 11 : Polling the value of *sysUpTime* to obtain the current system time

Step 12 : Polling *etherStatsOctets* (*Pkts*) to figure out the total traffic volume of a segment to be analyzed

Step 13 : Polling MIB variables

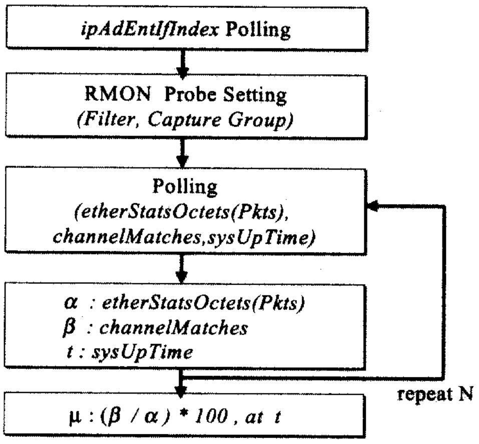
We perform polling the group of variables V{*captureBufferPacketData*, *captureBufferPacketLength*, *channelMatches*}.

Step 14 : Repeating the process from Step 11) to Step 13) during the polling frequency(n)

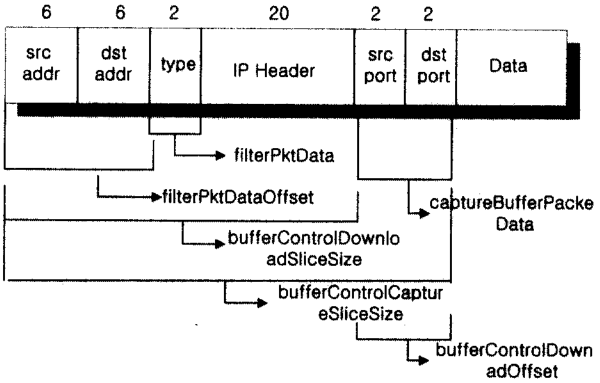
Step 15 : Setting the filter and the capture of the RMON probe as *invalid(4)*

In order to remove the filter and the capture from the RMON probe table, we set *filterStatus* and *channelStatus* of the filter group which has the index number I_{ro1} as *invalid(4)*, and we also set *bufferControlStatus* of the capture group which has the index number I_{ro2} as *invalid(4)*.

(Fig. 1) is a flow diagram of the analysis algorithm which is designed to get analyzing parameters by following the steps explained above. It shows that when the value of *etherStatsOctets* (*Pkts*) is set as α, and the value of *channelMatches* as β, the usage rate of the application service on the segment to be analyzed is μ.



(Fig. 1) Algorithm to get parameters analyzing an application on a LAN segment.



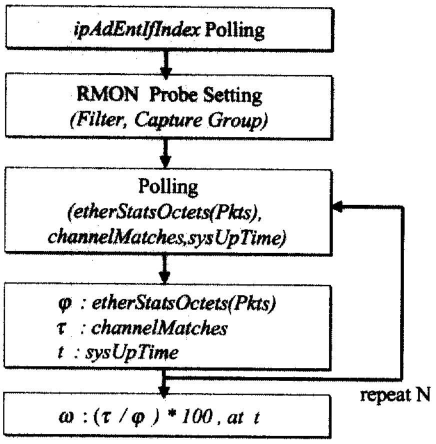
(Fig. 2) Applying RMON MIB to an Ethernet frame in order to analyze the application traffic on a LAN segment

(Fig. 2) shows how to apply objects of the filter group and the capture group to an Ethernet packet frame in order to analyze the application traffic. Analyzing the application traffic on a LAN segment allows us to classify all the packets according to the types of application services and to figure out what kinds of application services and how they are currently provided, thus understanding the characteristic of the traffic.

(2) Analysis of application services on a certain host in the LAN

The algorithm to analyze application services on a certain host is illustrated in (Fig. 3). It is similar to the one to analyze application services on a LAN segment. One small difference is that when we set the RMON MIB groups, we should set a certain host and a certain application service to be analyzed as the parts for filtering.

Step 1~Step 2 : Same as the analysis of application services on a LAN segment.



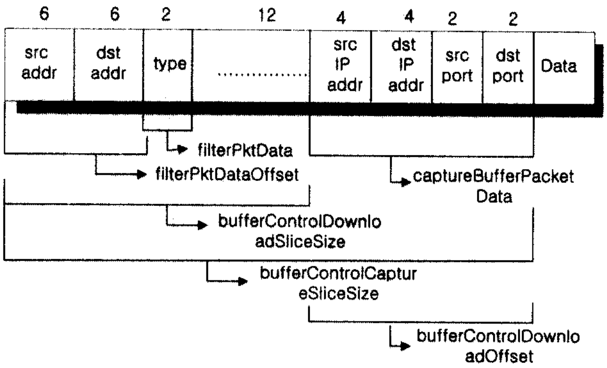
(Fig. 3) Algorithm to get parameters analyzing an application on a certain host

Step 3 : Setting the location and the value of a certain host and a certain application service for filtering

Using the index number I_{rol}, we set the values of a certain host and a certain application service to be filtered and analyzed at *filterPktData* of the filter group, and set *filterPktDataOffset* which designates the location in the Ethernet frame where we start filtering. We also set the values of OxFF and OxOO at *filterPktDataMask* and *filterPktDataNotMask* respectively which indicate the range of data which are set to match.

Step 4~Step 15 : Same as the analysis of application services on a LAN segment.

(Fig. 4) shows the way of filtering a certain host in the Ethernet frame. We perform filtering the source address and the destination address fields in the IP header of the Ethernet frame in order to filter a certain host to be analyzed, and we filter the source port and the destination port when it comes to filtering a certain application service to be analyzed. Analyzing the application traffic under a certain host allows



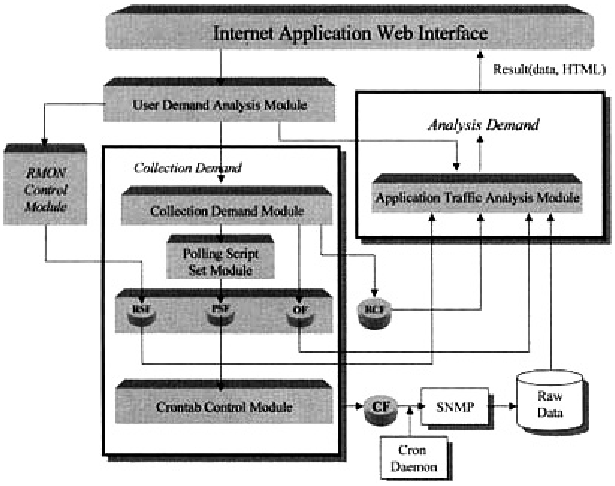
(Fig. 4) Applying RMON MIB to an Ethernet frame in order to analyze the application traffic on a certain host us to classify all the packets inflowing into and outflowing from the host on a LAN segment according to the types of application services, leading us to figure out what types of application services and how they are provided to users. Hence we can effectively recognize the characteristic of application services on the certain host in the LAN segment, as well as the characteristic of the traffic created by hosts on the LAN segment.

3. System Analyzing the Internet Application Traffic

In this paper, we develop a system determining and analyzing the traffic of Internet application in which polling frequency and polling interval are put at the disposal of network managers. (Fig. 5) illustrates its whole model and structure. This system is capable of analyzing the current situation and the usage rate of application services on a LAN segment or on a certain host in a LAN segment. Detailed explanations about each module are as follows :

3.1 Internet Application Web Interface

This is a web interface system between the system determining and analyzing the Internet application traffic and users, which delivers the users' demands to the analyzing system and sends the result of analysis from the system to the users.



(Fig. 5) Model and Structure of the System Determining and Analyzing the Internet Application Traffic

3.2 User Demand Analysis Module

It receives users' demands from the web interface, analyzes and sends them to an appropriate module to process them. It is divided into RMON Control Module, Collection Demand, and Analysis Demand to process users' demands real-time.

3.3 RMON Control Module

The RMON Control Module is a module which sets RMON according to the messages received from User Demand Analysis Module. It has two different procedures to set RMON : when it sets RMON at the request of the users to analyze data real-time, and when it sets RMON in the case of Collection demand. In order to set RMON, it reads and records the RMON data set in the RMON Set File (RSF), an environment file. (Fig. 6) shows the structure of RMON Control Module which is used to control RMON in the way mentioned above. The RMON Control Module is divided into rmon_valid, rmon_preset, rmon_invalid, and rmon_getip modules.

3.4 Collection Demand

Collection Demand is a module which collects data in order to analyze application services on a LAN segment. According to the range of management which request collection, it collects data of the managed system sequentially, and the data of the managed object simultaneously. It creates, modifies, or deletes Organization File (OF), an environment file which has data bout the users' collection demand. It is divided into Collection Demand Module, Polling Script set Module, and Crontab Control Module.

3.5 Analysis Demand

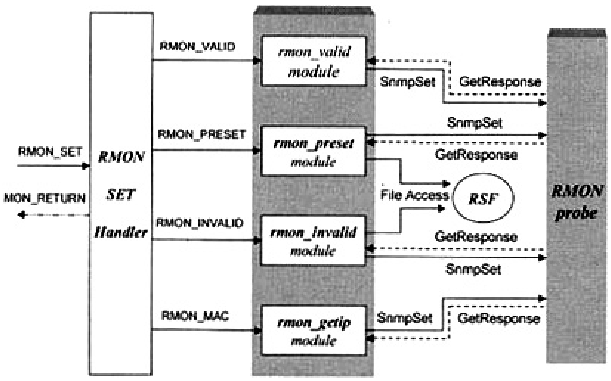
It inputs the file of management data collected by the Collection Demand during a specific period of time into Application Traffic Analysis Module which applies the analysis algorithm appropriate to analysis parameters to the data, and it delivers the result to the web interface.

3.6 Crontab Control Module

It registers Cron Daemon which is used for regular collection of data : it creates, modifies, deletes, or adds Crontab File (CF). Especially in this module can polling frequency be flexibly modified by the users.

3.7 Application Traffic Analysis Module

This module is activated in Analysis Demand. It analyzes the application service traffic created on the network and applies an analysis algorithm to it according to an appropriate analysis parameter.



(Fig. 6) The Structure of RMON Set Module

4. Experiments and Analysis

Using the methodology, the algorithms and the system designed to determine the Internet application traffic, we in this paper conducted experiments on the network of our University to which we belong. The LAN segment used for the experiments was the 203.252.53.0 network, and the address of the RMON probe was 203.252.53.57 which uses 9 groups. The host analyzed in these experiments was the 203.252.53.41 system. In order to prove the efficiency of the algorithm we suggest in this paper, we installed *Baystack* hub which is equipped with SA NMM, RMON agent produced by Bay Co., on a specific network (203.252.53.0). Using this hub, we collected defined management data and performed calculation according to the algorithm.

[Experiment Environment]

(1) We conducted the experiments on some designated segments of the LAN of our University at the speed of 10Mbps.

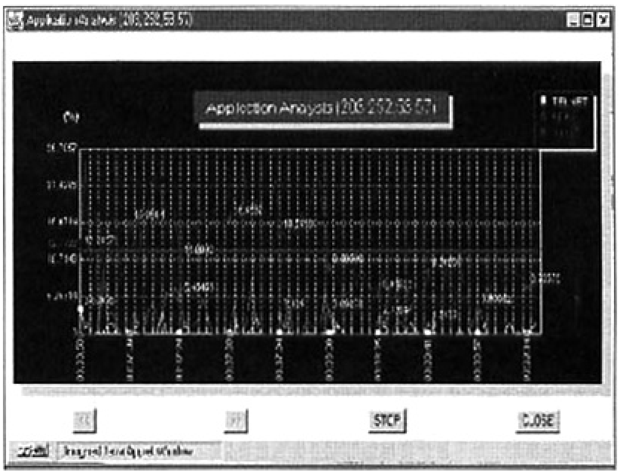
(2) Using the hub equipped with the RMON agent (203.252.53.57) existing on the LAN segments, we performed polling regularly and collected management data for two months starting from April 14, 1999 to June 13, 1999.

(3) For regular monitoring, we used crontab on Solaris 2.5, and the interval of monitoring was 10 minutes.

Our experiments were conducted under the two scenarios as follows :

4.1 when the network works normally

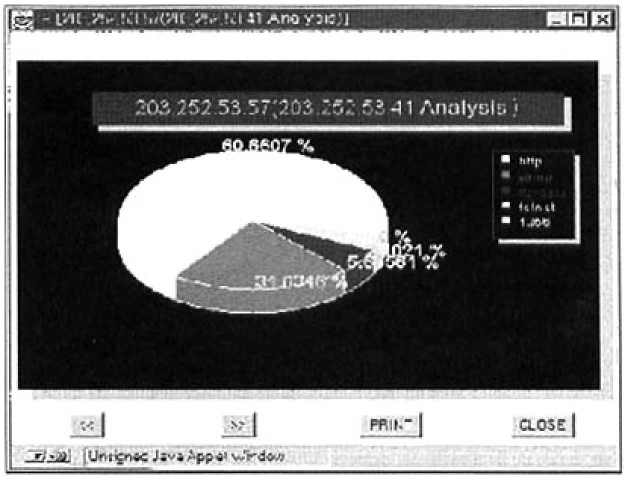
In the situation when the network worked normally, we analyzed the application traffic flowing on the LAN segment and the traffic flowing under the specific host. (Fig. 7) indicates the usage rates of TCP application services including Telnet, ftp, and http on the LAN segment, and (Fig. 8) shows the types and the usage rates of the application services under the host (203.252.53.41) which were used during a designated period of time. This experiment shows that the http protocol is used the most among the application services, followed by snmp and ftp.



(Fig. 7) Analysis of TCP services (Telnet, ftp, http) on the LAN segment

4.2 when the network works abnormally

The second scenario we paid attention to was when the network came to not operate because of the excessive traffic on the LAN segment generated by broadcasting. We analyzed the application traffic under the certain host to find out the cause of the problem, and controlled it. As (Fig. 9) indicates, the network came to work abnormally, because the host (203.252.53.10), working as a mail-server generated SMTP messages excessively.


203.252.53.57(203.252.53.41 Analysis)
60.6607 %
31.0346 %
5.6581 %
http
telnet
PRINT
CLOSE
Unsigned Java Applet Window

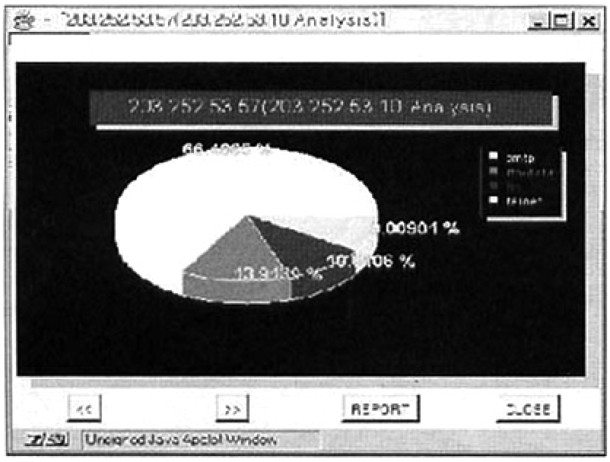
(Fig. 8) Analysis of application services under the certain host (203.252.53.41)

<Table 1>, as well as (Fig. 8), shows the result of analysis under the first scenario.

<Table 1> Analysis of application services under the certain host in a normal network condition

Factor	Analysis Parameter	RMON IP Address (203.252.53.57)				
		Address of the host analyzed (203.252.53.41)				
Host	Applications	HTTP	SNMP	FTP-data	Telnet	1256
		60.66%	31.63%	5.6%	2.1%	0.2%

According to <Table 1>, among the application services, HTTP accounts for the biggest part of the traffic, more than 60%, followed by SNMP and FTP. It shows that the host analyzed here acts as a Web server. In our experiments for this paper, we monitored the traffic to be managed and, when an unexpected problem occurred, figured out its cause and a solution to it, using the methodology, the algorithms, and the system we presented in the first part of this paper.



(Fig. 9) Analysis of the application traffic under a specific host when the network works abnormally

<Table 2>, as well as (Fig. 9), shows the result of analysis under the second scenario.

<Table 2> Analysis of application services under the certain host in an abnormal network condition

Factor	Analysis Parameter	RMON IP Address (203.252.53.57)			
		Address of the host analyzed (203.252.53.10)			
Host	Applications	SMTP	FTP-data	FTP	Telnet
		66.46%	13.91%	10.61%	9%

According to <Table 2>, the SMTP traffic accounts for more than 66% under the analyzed host of 203.252.53.10, which allows us to assume that the mail protocol of the 203.252.53.0 segment and the SMTP protocol of the FTP server (203.252.53.10) are the cause of serious system errors affecting all the LAN segments. After further scrutiny, we realized that the sendmail of the server was not properly set up, continuously generating ineffective packets.

RMON probe devices used to analyze the application service traffics in this paper allow us to collect and analyze the traffics more accurately and quickly than other ways of analyzing the traffics using a software, and thus to provide the users with the application service traffics on the LAN more accurately. The algorithm we suggest in this paper is the only way of analyzing the application service traffics when the 9 groups of RMON probe devices are used. Furthermore, the algorithm makes it possible to monitor the application services in the troubled system, to determine which service causes the system error, and to seek a quicker and effective solution to it.

5. Conclusion

Due to the rapid development of the Internet and the surge of its users, Internet application services have been increased and diversified, so as to account for most of the network resources. In order to ensure the supply of reliable and effective Internet application services in this situation, their traffics have to be monitored and managed. Unfortunately, however, international standards for managing Internet application services are not mapped out yet, which forces network managers to rely on the existent frameworks and technologies for the overall network management. Taking this reality into account, we decided to develop a new system which is designed to extract the Internet application traffic among the whole bunch of the traffics created on a network. In this paper, we used RMON devices working on the network layers in order to determine the Internet application service traffic, defined analysis parameters and suggested analysis algorithms for the management of the services. We also created and activated a system which extracts and analyzes the Internet application traffic on a certain network, and, based on the data, we developed an integrated manager system designed to manage Internet application services. The Integrated Internet Application Service Manager System we developed in this paper uses Java workable on all web interfaces and networks, allowing us to manage all application services beyond the limitation of time and space. In conclusion, the system has made it possible to manage all the application services and furthermore, the application traffic on each segment.

References

[1] Ahn Sung Jin, "*Algorithms to Calculate System Analysis Parameters for TCP/IP Network Management*," Sungkyunkwan University, Ph D thesis, 1998.

[2] Peter B.Danzig.Katia Obraczka, Anant Kumar, "*An Analysis of Wide Area Name Server Traffic*," August, 1992.

[3] J. P. Thompson, "*Web-Based Enterprise Management Architecture*," IEEE Communication Magazine, March 1998.

[4] B. Reed, M. Peercy and E. Robinson, "*Distributed Systems Management on the Web*," in Proceedings of the 1997 IEEE Integrated Management, San Diego, CA, 1997.

[5] C. Kalbfleisch, C. Krupczak, R. Presuhn, J. Saperia : *Applications Management MIB*, Internet Draft<draft-ietf-applmib-mib-00.txt>, November, 1996.

[6] H. Hazewinkel, *Survey of Defined Managed Objects for Applications Management*. Internet Draft <draft-hazewinkel-appl-mib-00.txt>, January, 1997.

[7] Thomas Johannsen, Glenn Mansfield, "*A Study of FTP Traffic Estimation of the NonOptimality*," December, 1994.

[8] C. Anthony Cooper, Robert J. Schmidt, "*Performance Characterization and Assessment for Internet Services*," Networld Interop'98, May, 1998.

[9] Takayuki Kushida, "*The Traffic Measurement and the Empirical Studies for the Internet*," GLOBECOM'98, Vol.2, 1998.

[10] Nathan J. Muller, "*Web-accessible Network Management Tools*," International Journal of Network Management Vol. 7, Wiley, pp.288-297, 1997.

[11] Jonas Andren, Magnus Hilding, Darryl Veitch, "*Understanding end-to-end Internet Traffic Dynamics*," GLOBECOM'98, Vol.2, 1998.

[12] RFC 1757, "*Remote Network Monitoring Management Information Base*," S. Waldbusser, February, 1995.

[13] M. Toet, "*A Prototype for World Wide Web Management*," M. Sc. Thesis Enschede, April, 1997.

[14] Nathan Kalowski, "*Applying the RMON Standard to Switched Environments*," International Journal of Network Management Vol.7, Wiley, 1997.

[15] J. Hong et. El., "*Web-based Intranet Services and Network Management*," IEEE Communications Magazine, October, 1997.

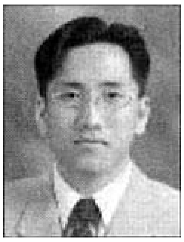
[16] William Stallings, "*SNMP, SNMPv2 and RMON*," Addison-Wesley Publishing Company, 1996.

[17] Jeong-Soo Han, Seong-Jin Ahn, Jin-Wook Chung, "Web-based Performance Manager System for a Web Server," NOMS'98, 1998.

[18] J. Judge, H. W. P. Beadle, J. Chicharo, "Sampling HTTP Response Packets for Prediction of Web Traffic Volume Statistics," GLOBECOM'98 Vol.5, 1998.

[19] Kang Hong Cho, Sung Jin Ahn, Jin Wook Jung, "Algorithms for Calculating LAN Performance Parameters using RMON MIB," KICS'99 Vol.24, 1999.

[20] Kang Hong Cho, Sung Jin Ahn, Jin Wook Jung, "Rule-based Fault Detection Agent System for Fault Detection and Location on LAN," KIPS'2000 Vol.7, 2000.



한 정 수

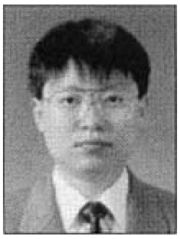
e-mail : jshan@songgang.skku.ac.kr

1997년 성균관대학교 공과대학 정보공학과 졸업(학사)

1999년 성균관대학교 전기전자 및 컴퓨터공학부 석사 졸업

현재 성균관대학교 전기전자 및 컴퓨터공학부 박사수료

관심분야 : 네트워크 관리, 트래픽 분석, 인터넷 QoS, QoS 라우팅



안 성 진

e-mail : sjahn@comedu.skku.ac.kr

1988년 성균관대학교 정보공학과 졸업(학사)

1990년 성균관대학교 대학원 정보공학과 졸업(석사)

1990년~1995년 한국전자통신연구원 연구전산망 개발실 연구원

1996년 정보통신 기술사 자격 획득

1998년 성균관대학교 대학원 정보공학과 졸업(박사)

1999년~현재 성균관대학교 사범대 컴퓨터교육과 조교수

관심분야 : 네트워크 관리, 트래픽 분석, Unix 네트워크

정 진 욱

e-mail : jwchung@songgang.skku.ac.kr

1974년 성균관대학교 전기공학과 학사

1979년 성균관대학교 대학원 전자공학과 석사

1991년 서울대학교 대학원 계산통계학과 박사

1982년~1985년 한국과학기술 연구소 실장

1981년~1982년 Racal Milgo Co. 객원연구원

1985년~현재 성균관대학교 전기전자 및 컴퓨터공학부 교수

관심분야 : 컴퓨터 네트워크, 네트워크 관리, 네트워크 보안