ELSEVIER

Computer Communications 22 (1999) 1333–1342

computer communications

WebTrafMon: Web-based Internet/Intranet network traffic monitoring and analysis system

J.W.-K. Hong*, S.-S. Kwon, J.-Y. Kim

Department of Computer Science Engineering, Pohang University of Science and Technology, San 31 Hyoja, Pohang 790 784, South Korea

Abstract

As enterprise computing environments become more network-oriented, the importance of network traffic monitoring and analysis intensifies. Most existing traffic monitoring and analysis tools focus on measuring the traffic loads of individual network segments. Further, they typically have complicated user interfaces. As Internet and Intranet traffic increases due to the increase in the use of the World-Wide Web and other applications, determining which host and which application generates how much network traffic is becoming critical in managing and using network resources effectively. This paper presents the design and implementation of a portable, Web-based network traffic monitoring and analysis system called WebTrafMon. Web-based technology enables users to be free from complex user interfaces, while allowing monitoring and analysis results to be viewed from any site, using widely available Web browsers. WebTrafMon provides monitoring and analysis capabilities not only for traffic loads but also for traffic types, sources and destinations. WebTrafMon consists of two parts: a probe and a viewer. The probe extracts raw traffic information from the network, and the viewer provides the user with analyzed traffic information via Web browsers. The effectiveness of WebTrafMon has been verified by applying it to an enterprise network environment.

Keywords: Network traffic monitoring and analysis; Traffic management; Traffic type and source analysis; Web-based management; Enterprise network management

1. Introduction

Today, more and more, enterprise computing relies increasingly on computer networks. Network environments are growing, fueled by extensive enhancements of computer hardware and software as well as the rapid growth of the Internet and World-Wide Web (WWW or the Web). More systems are being connected to networks, rapidly increasing traffic. Such growth impacts on the performance of many network-related user applications. These conditions gave birth to various problems and solutions for network traffic monitoring and analysis.

Network traffic monitoring provides collected data and analysis of this data. Raw status data is gained by probing network packets, and analysis provides extended data based on the raw data. To use limited network resources effectively, network managers need to obtain accurate and reliable information from their networks, such as:

- how much traffic is transferred;
- what type of traffic is transferred;
- how much traffic is generated from which system;
- which system or application is causing bottlenecks; and
- how high is peak traffic and when does it peak.

If network managers cannot provide reliable answers to these kinds of questions, valuable network resources will be wasted.

A number of automated tools have been developed in order to help network managers monitor and analyze network traffic. Multi-Router Traffic Grapher (MRTG) [1], Argus [2], Etherfind [3], NFSwatch [4] and TCPdump [5] are examples of such tools. MRTG provides hourly, daily, weekly, monthly and yearly statistics of network traffic loads using Simple Network Management Protocol (SNMP) [6] through the use of Web-based graphical user interfaces. MRTG, unfortunately, does not provide details concerning protocols used or system traffic origin. Although MRTG is an excellent tool with easy-to-use Web-based user interfaces, network traffic managers need more sophisticated and user-friendly tools to analyze detailed traffic information.

This paper presents the design and implementation of a new Web-based network traffic monitoring and analysis system called WebTrafMon. The proposed system shows network traffic in detail, allowing users to see statistics at

* Corresponding author. Tel.: + 82-562-279-2244; fax: + 82-562-279-5663.

E-mail address: jwkhong@postech.ac.kr (J.W.-K. Hong)

0140-3664/99/$ - see front matter
PII: S0140-3664(99)00130-9

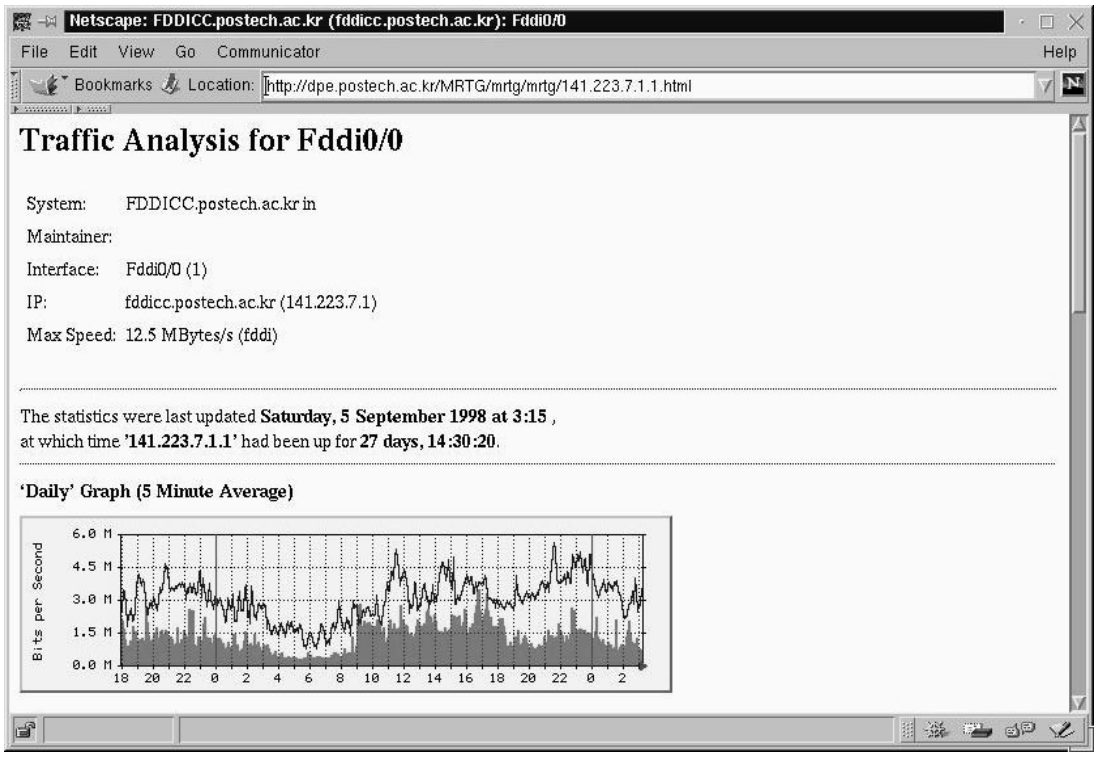

Fig. 1. Traffic load statistics of a network segment using MRTG.

each communications layer, such as source and destination, total number of packets transferred, and other data. WebTrafMon employs Web interfaces so users can access traffic information via generic, easy-to-use and ubiquitous Web browsers. Web interfaces are system and location-independent and the most generic way of showing what WebTrafMon can do. Users only need to be connected to the Internet and have a Web browser. No special training is required. No additional program installation or configuration is needed on the client side. Anyone can access the data and analysis, although we can restrict access by using a common Web-based user authentication.

WebTrafMon extracts information using network packets from headers at the lower Medium Access Control (MAC) level, to that of the upper application levels. No additional network information exchange or overhead is needed to obtain the information. The information is extracted in real time, so users can see the network's activities, and which system or protocol is becoming dominant—from the very first time they use WebTrafMon.

WebTrafMon differs from similar tools because it can show traffic information according to the source and destination host through any Web interface; and it can show the traffic status according to each protocol layer, from the network layer to the application layer. Often when networks are heavily loaded, traffic originates from a number of specific hosts. Network managers of enterprise network environments should be able to ascertain quickly which host is making a bandwidth peak. Once identified, the responsible host causing the bottleneck, can be contacted via the system administrator who then corrects the situation.

Traditional network traffic analysis tools are unable to show traffic per host. Traffic information per host, however, nowadays is important data. For example, only one unauthorized Video On Demand (VOD) or Audio On Demand (AOD) server can cause an entire subnet to peak. More and more multimedia services are being offered on the Internet, some consuming a great deal of bandwidth. Thus information about protocols from each level is vital, but also, host traffic data becomes more and more critical.

WebTrafMon can be divided into two parts: a probe and a viewer. The probe extracts data from network packets and composes log files. Analysis results are based on these log files. The viewer interacts with users, showing the analysis results to the user through Web browsers. What follows is the detailed description and internal structure of our system. Section 2 examines other network traffic monitoring solutions. Section 3 examines the requirements for traffic monitoring and analysis systems. Section 4 presents the design architecture of WebTrafMon. Section 5 describes information specifications used to build WebTrafMon. Section 6 examines the use of this system for analyzing enterprise networks. Finally, Section 7 concludes this paper and examines future research.

2. Related work

Network monitoring and analysis have become important topics in networking environments, encouraging worldwide research and development. Many organizations have developed automated network monitoring and analysis systems, some of which are described below.

2.1. Multi-Router Traffic Grapher

The MRTG [1] is a tool for monitoring traffic loads on network-links. MRTG generates HyperText Markup Language (HTML) pages containing GIF images that provide a live, visual representation of network traffic. MRTG is implemented using Perl and C languages and it can be operated under various UNIX platforms and Windows NT. MRTG is successfully used on many sites around the world.

MRTG is not just limited to monitoring network traffic. It can be used to monitor any SNMP MIB [6] variable. MRTG can even supply analysis results using data gathered from an external program. People are using MRTG to monitor such information as system load, login sessions, modem availability and more. MRTG even accommodates two or more data sources into a single graph. Fig. 1 shows a screen shot of network load monitoring using MRTG.

Despite all these attractive capabilities, MRTG cannot provide information that shows which host or application may be causing a traffic bottleneck. SNMP MIB variables are not appropriate for such use and, for traffic, it can only show traffic load. MRTG does not provide information about traffic type or protocol statistics.

2.2. Etherfind

The SunOS operating system provides Etherfind [3], which is a software packet monitor. The software opens the network card in the promiscuous mode and writes a summary line of each packet to a file. Information includes protocol type, size, and sending and receiving addresses. This tool extracts information from each packet. Data is supplied as a text-based user interface, and only users with root permission can access the tool.

2.3. NFSwatch

NFSwatch [17] monitors all incoming network traffic destined to NFS file servers, and divides it into several categories. The number and percentage of packets received in each category is displayed on the screen, which is continuously updated. By default, NFSwatch monitors all packets destined for the current host. An alternate destination host to watch for may be specified using an internal argument. If a source host is specified with the -src argument, then only packets arriving at the destination host which were sent by the source host are monitored. This tool, like Etherfind, is inappropriate for our needs because it was originally designed to monitor a single host.

2.4. TCPdump

Originally written by Van Jacobson, TCPdump [5] was developed to research and improve Transmission Control Protocol (TCP) and Internet gateway performance. Recent versions make use of the system-independent packet capture library (libpcap) [8]. TCPdump prints the headers of packets on a network interface. And the users analyze network status using this header information manually. Though TCPdump has many options for capturing raw data, it does not provide any analysis capability of the captured data.

2.5. Argus

Argus [2] is a generic Internet Protocol (IP) network transaction auditing tool that has been used by many sites worldwide, performing many powerful network management tasks. It runs as an application level daemon, promiscuously reading network packets from a specified interface. It then generates network traffic audit records for the network activity that it encounters. The manner in which Argus categorizes and reports on network activity makes this tool unique and powerful.

First, it extracts information from each packet in the promiscuous mode and saves the information to a file and later analyzes that file. This tool skillfully shows information about protocols, but does not show source or destination host information. Further, it only provides a text-based user interface.

2.6. Etherload

Etherload [9] is a freely available LAN traffic analyzer for MS-DOS system with an Ethernet or Token Ring controller. As the name of the program represents, this program was originally developed for measuring traffic load of Ethernet segments. Etherload basically captures each packet running through the LAN and provides various information on the packet. The available information covers a number of network protocols such as TCP/IP, DECnet, OSI network, Novell's Netware network, Microsoft's NetBEUI network, etc. The program displays important parameters, events and loads for the protocols as well as the simple overall load statistics.

Etherload can be used to check which host is generating the most traffic, which host is sending to which host, and what kind of protocols are in use in a specific Ethernet segment. It provides detailed parameter information and statistical data of the segment traffic. Especially for the TCP/IP network, Etherload is able to analyze and classify traffic by TCP/UDP port numbers, which give statistical information for various TCP/IP applications. Since Etherload is based on the MS-DOS environment, it provides a character-based user interface for displaying traffic

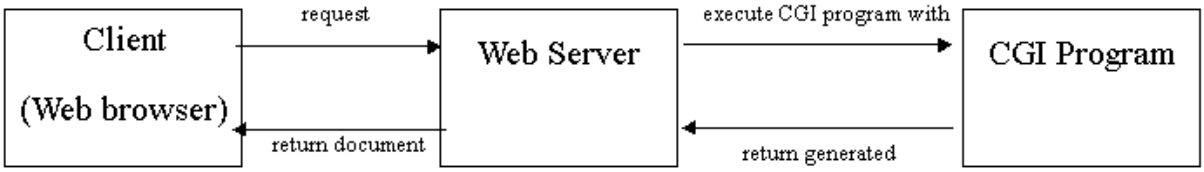

Fig. 2. Principle architecture of a common gateway interface.

information. Each protocol information is displayed on a separate screen and users can switch the screens by typing commands.

After a careful examination of the above-mentioned tools, it was found that none of them provided the capability required for accurate network monitoring and analysis. This motivated the development of WebTrafMon.

2.7. Web technology

The WWW [10] was developed originally by researchers at CERN. The Web is a way of organizing and accessing online content in multiple media using hyperlinked relations on the Internet. The native protocol of the Web is HyperText Transfer Protocol (HTTP) [11], used on the Web since 1990. HTTP is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, object-oriented protocol that uses an extension of its request methods (commands) for name servers and distributed object management systems. HTML is the simple data format used to create hypertext documents that are portable from one platform to another.

The Web is based on a client/server architecture and basically operates as follows. Hypertext documents that are to be made available on the Web are prepared in HTML and made accessible by the Web server. Users wishing to retrieve the documents can do so by using a Web client (Web browsers such as Netscape Navigator or Internet Explorer) to connect to the Web server which contains the documents.

The Common Gateway Interface (CGI) [12] is the most popular method for interfacing external applications with Web servers. An HTML document, that the Web daemon retrieves, is static, which means that it exists in a constant state: a text file that does not change. A CGI program, however, is executed in real-time, so that it can output dynamic information. If a person wants to hook up their database to the Web so people worldwide may query it, then the person needs a CGI program. The CGI program must allow the Web server to transmit information to the database engine, receive the results back again, and display them to the client.

Fig. 2 illustrates the architecture of the Web with a CGI. Using CGIs, what can be hooked up to the Web is practically limitless. Many currently available Web-based management solutions use CGIs as gateways to management agents.

3. Requirements

The following are the most important requirements that should be satisfied in an enterprise network monitoring and analysis system.

3.1. Platform independence

Low-level packet capturing routines should be platform-independent. Because each platform provides a different low-level network device, there should be an abstraction above the base network layer. If the code is written for one specific platform, porting the program to different platforms would be hard, and users of those platforms will not be able to use it. For Unix, the BSD networking code is the common basis of each proprietary operating system. Thus, packet-capturing routines can be based on a common code. Porting this program to a PC Windows platform would require additional work because on low level network devices the Unix platform and Windows platform have almost nothing in common.

3.2. Powerful user interfaces

User interface factors should be easily understood and manipulated. For this reason, a Web based interface may be the best solution. Web-based user interfaces are easy to use and ubiquitous because the Web is not dependent on a specific operating system. As well, Web browsers are available for most operating systems.

A Web-based user interface has one more specific strong point. To use a Web-based system, a person only needs access to the network and use of a Web browser. Anywhere, anytime, anyone can access the system using a universal Web browser.

3.3. Guaranteed packet capturing

On a high-speed network, the number of packets transmitted per second can be astronomical. Analyzing such data requires a great deal of processing time. Thus, the efficiency of a packet capturing code is essential.

On a high-speed network, the system may not be able to

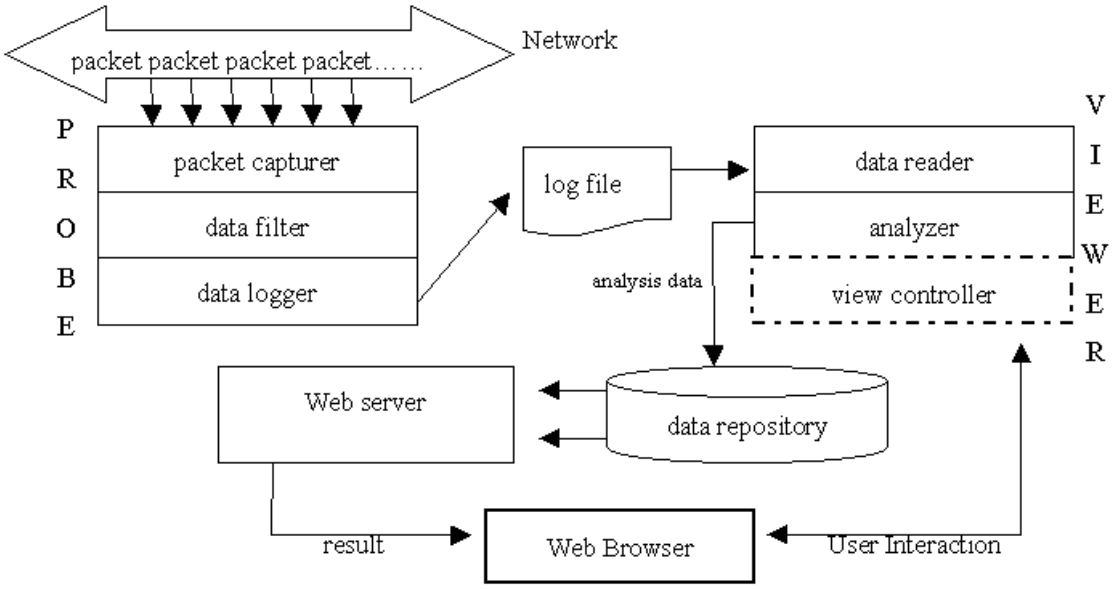

Fig. 3. Design architecture of a Web-based traffic monitoring & analysis system.

handle all the packets in time. If the processing speed, or the system itself, acts at speed insufficient to capture all packets, the analysis result is unreliable. As a matter of fact, ensuring that all packets are properly captured is extremely challenging [13].

3.4. Classification of all possible protocol information

Many communications protocols are currently used on a typical enterprise network. For example, numerous application protocols exist, such as HTTP, FTP, Telnet, SNMP, MP3, RealAudio, RealVideo, etc. All packets are delivered using some specific protocol, and the protocol can be classified into a certain layer. Ideally, a monitoring tool should be able to classify and display all possible protocols in each layer.

3.5. Mobility

A packet-capturing tool should be easy to install and use on any network segment. If someone wants to monitor a specific network segment, he should be able to install the monitoring system on a notebook or desktop and connect it to that segment easily.

3.6. Security

Security is vital. Securing internal data is necessary to prevent illegal access and potential damage to data. Sometimes, pranksters or hackers crack systems that neglect security measures. Access must then be restricted to those users who are authorized. Accordingly, a Web security mechanism (such as username and password checking) can be used to provide such access to the monitoring and analysis system.

3.7. Viewing of real-time and historical data

The system should be able to show the online real-time status data and accumulated historical status data easily. From the historical data, the user can analyze long-term traffic trends and from the real-time data, short-term traffic trends. This helps the user to detect problems easier and faster.

4. Design of WebTrafMon

Based on the requirements discussed in the previous section, we have designed a Web-based enterprise network monitoring and analysis system. The design architecture of WebTrafMon is illustrated in Fig. 3.

The overall system consists of two parts: a probe and a viewer. The probe extracts data from network packets in each layer and saves it to a pre-configured log file. The log file is then processed by the analyzer component of the viewer. The user interacts with the output analysis data stored in the data repository, which the Web server sends to the user's Web browser through the view controller.

4.1. Probe

The probe is used to retrieve data from each captured packet and store the data in a log file for further processing. To capture all network packets, it acts in the promiscuous mode. This is due to the property of the Ethernet itself: packet broadcasting. Capturing all packets is the most

Fig. 4. Medium access layer packet.

important and basic operation of the probe. Therefore it should be implemented using an operating system independent code. A detailed description of the structure of the probe for each data communications layer follows.

But first, here is the role of each component of the probe. The packet capturer captures packets from the network. The data filter extracts the information of each packet. The data logger writes the information that the data filter extracted to the log file.

4.1.1. Medium Access Control layer

MAC layer packets should be captured at full speed. However, it is practically impossible to capture all packets reliably because this is not a hardware solution but a software solution, of which the packet capturing routine should be concise and efficient. Fig. 4 shows the packet structure of the MAC level [14].

At this layer, the abstraction of some specific hardware drivers (e.g. Ethernet card) is important. If the packet capturing code uses specific network device drivers, porting this code to another network device may require additional work.

The probe extracts all traffic-related information from this layer and the size of each packet from its header.

4.1.2. Network layer

This layer is related to IP, Address Resolution Protocol (ARP) [14], Reverse ARP (RARP) [14], and so on. If a packet is IP-based, it is further processed to analyze the source and destination host information. If not, there is no need to analyze it because it does not contain the source or destination host information. The viewer must then be informed that the packet is not IP-based so it knows that the destination or host information is not available.

As for determining which host sent a packet, information from this layer is used. A standard IP packet has two IP addresses in it: source and destination. The system uses this valuable information to analyze per-host traffic [14].

4.1.3. Transport layer

This layer is the host-to-host transport layer. The two most prevalent protocols in the transport layer are TCP [7] and User Datagram Protocol (UDP) [7]. TCP provides reliable data delivery service with end-to-end error detection and correction. UDP provides low-overhead, connectionless, best-effort datagram delivery service.

The probe extracts the protocol that was used at this level for any given packet, and saves the information to the log file for further use by the viewer.

4.1.4. Application layer

The application layer is directly related to the source and destination ports, providing information about the application protocols and destination port contained in the packet [7]. For example, port number 23 is used by Telnet. This port number is reserved for Telnet use only. Any port number lower than 1023 has been reserved. The RFC 1700 [15] defines the usage of the ports below 1023 and these are called “Assigned Numbers”. From this data, we can resolve the packet and know what application level protocol has been used.

A large number of application layer protocols exist. Among them, HTTP related traffic has increased due to the popularity of the WWW. HTTP is a Multipurpose Internet Mail Exchange (MIME) [7] based protocol. Any data that can be defined using MIME can be transferred through HTTP [11,16]. This gave birth to multimedia Web services, which typically require an excess of bandwidth. Such services have attracted a lot of users to the Internet and hence, increased network traffic (both Internet and Intranet) has substantially in recent years. New Internet applications (and thus net application protocols) are being introduced to the Internet rapidly these days. A network monitoring system should be able to detect such new application protocols and analyze them.

4.2. Viewer

The WebTrafMon viewer consists of three components: data reader, analyzer and view controller. The data reader reads packet information from the log file that the probe has generated. The analyzer analyzes information that the view controller requested. Finally, the view controller interacts with the user to provide the information the user requests.

The viewer analyzes the log file generated by the probe and shows all possible information. This includes information on the protocols being used on the network, but it mainly concerns the network bandwidth use of:

- each network layer protocol;
- each transport layer protocol;
- each application layer protocol;
- traffic between each source and destination;
- traffic from each source host;
- traffic to each destination host.

For ease and efficiency, a Web-based user interface has been used, although any interface can be used. A Web-based interface eliminates problems of porting, while a single

Table 1
Sample log file

346	164.124.96.18	141.223.82.4	udp	telnet
64	141.223.82.4	141.223.82.26	tcp	http
112	rarp			
64	arp			
74	141.223.99.99	141.223.82.28	icmp	

script provides uniform results, regardless of the operating system, wherever the user is located.

The viewer reads the log file generated by the probe and interacts with the user to determine what the user wants. Depending on the user interaction, the viewer processes the packet data from the log file and shows the result to the user through a Web browser.

In designing the viewer, all possible security problems have been considered. This demands that WebTrafMon authenticate users so that only those with permission can access and use the system. Password checking is the most general way of authentication. If password checking is too frequent though, the user may regard it as a nuisance. Therefore, checking passwords at the initial login stage is enough.

5. Implementation

This section presents implementation details of WebTrafMon.

5.1. Probe implementation

The packet-capturing component using libpcap [8], a system-independent interface for user-level network packet capturing developed by the Network Research Group at the Lawrence Berkeley Laboratory was implemented. The general libpcap API makes porting to, and support of multi-vendor systems easy. The libpcap interface supports a filtering mechanism based on the architecture of the BSD packet filter (BPF) [8]. Although most packet capture interfaces require in-kernel filtering, libpcap uses in-kernel filtering only for the BPF interface. Systems lacking BPFs use all packets that are read into user space. Then the BPF filters are evaluated in the libpcap library. BPF is standard in 4.4BSD, BSD/386, NetBSD, and FreeBSD. DEC OSF/1 uses the packet filter interface but has been extended to accept BPF filters (which libpcap utilizes).

To analyze packet information, a log format was defined according to the TCP/IP protocol layer and the size of each packet. Table 1 is a sample part of a log file that we store. The significance of each row and column is as follows. Each row represents a captured packet, with the first field showing the size, in bytes, of each packet. This information is extracted from the MAC layer. As for Ethernet packets, this information appears under the Ethernet header.

The second field is the source host of the packet, if it is an IP-based packet. If it is not based on IP, it may be based on ARP [14], RARP [14], or other protocols in this layer. For example, lines 3 and 4 from the sample are such cases. This information is extracted from the network layer.

The third field is the destination host of the packet if it is IP-based. The source and destination host information go hand-in-hand. This information is very important in order to learn from which host the traffic originated.

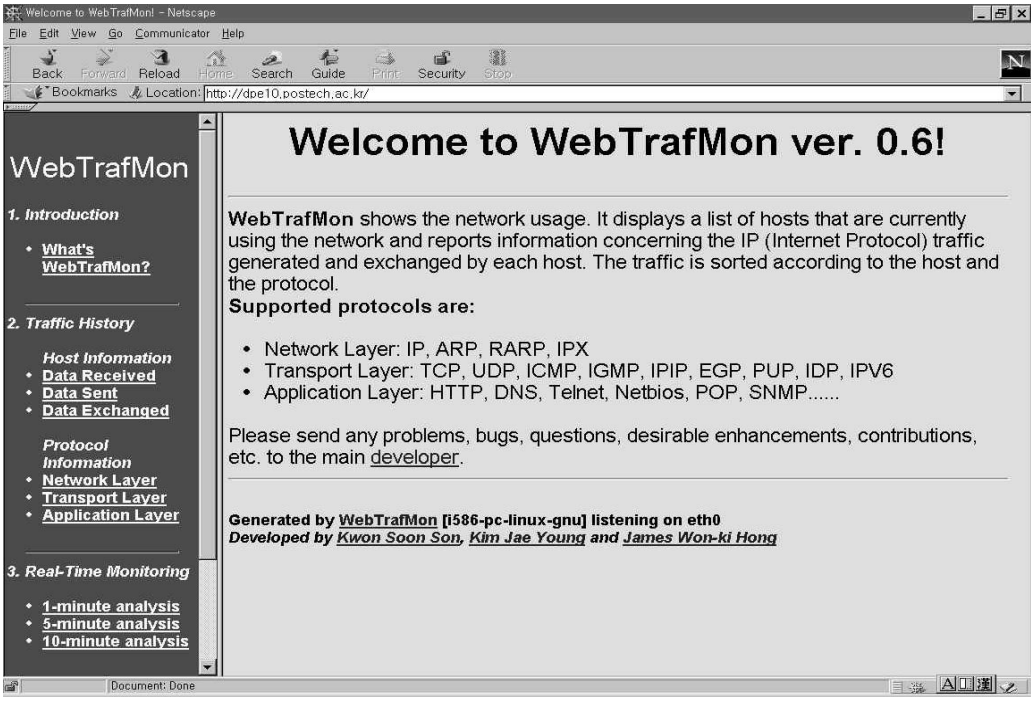

Fig. 5. WebTrafMon-analysis homepage.

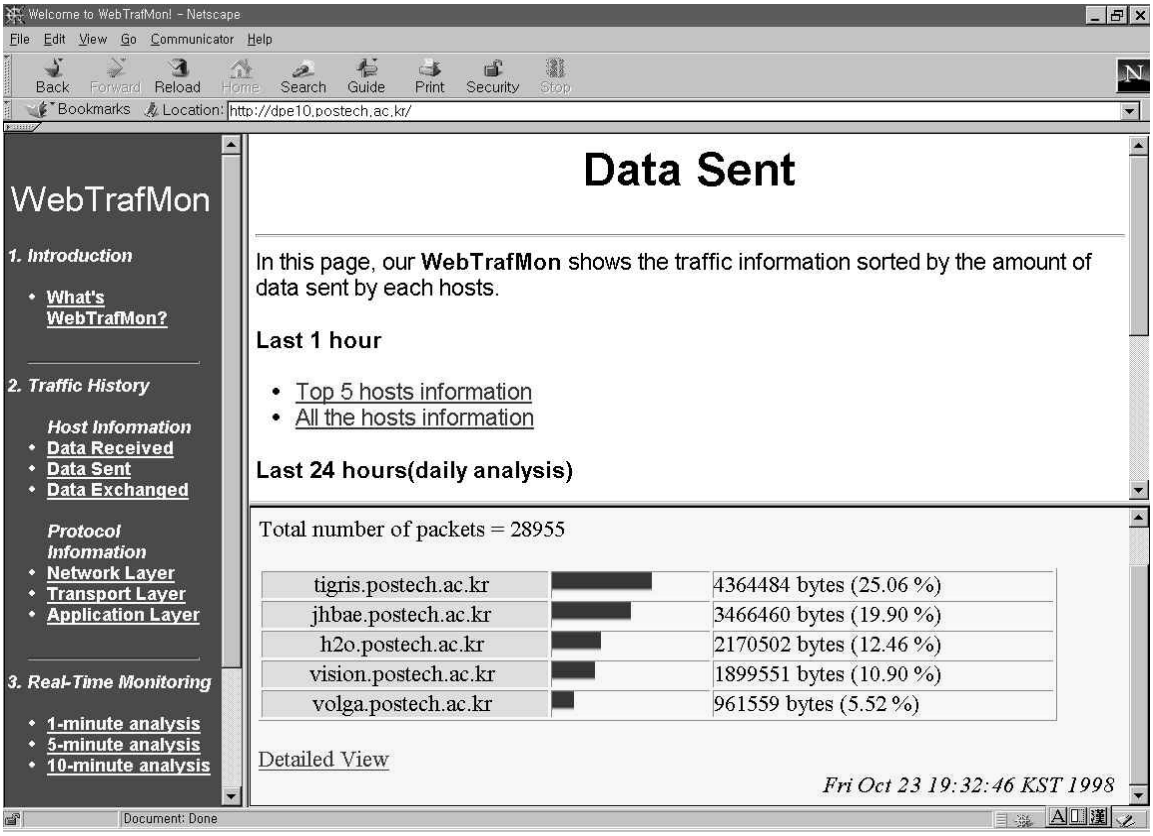

Fig. 6. Source host traffic analysis view.

The fourth field is the transport layer protocol information. If it is TCP-based, the flag “tcp” is printed; and if it is UDP-based, the flag “udp” is printed. If it is not based on TCP or UDP, the appropriate protocol name is printed. The last line is such a case.

The last field of each TCP or UDP-based packet is the application layer information. This is determined using the port number information of each TCP or UDP-based packet.

5.2. Viewer implementation

The viewer analyzes the log files generated by the probe. The Perl script was used to analyze the log file, and CGI to show the final output on the user’s browser. For each column, the Perl script retrieves information on each network layer and shows it graphically. All scripts gather source-host information, destination-host or protocol-related information, and packet size information. Showing such data is fairly simple. All that is needed is the addition of the HTML code. WebTrafMon can show the following traffic information for a given log file:

- source host;
- destination host;
- source to destination pair;
- network layer protocol;
- transport layer protocol;
- application layer protocol.

The viewer extracts one or two of the fields that is actually needed from the log file, and sorts it according to the total traffic. The traffic and bandwidth related information originates from the first field of the log file, the size of a given packet.

For security, we added a password-checking interface at the initial stage to authenticate the user. This can be done through the viewer component itself or by modifying the Web server side. It was determined that modifying the configuration of the Web server would be easier. The Apache Web server [18] that is used provides a simple and easy password authentication mechanism.

6. Experience

WebTrafMon is used on a demand basis. As long as they have security access, anyone who feels that the network has peaked or has some problems, can connect to our homepage to interact with this tool. The analysis can be done in real time or in traffic history. Fig. 5 shows the main window of WebTrafMon.

The left window shows what information WebTrafMon can display for the user. “Data Received” displays the destination host information. “Data Sent” displays the source host information. “Data Exchanged” displays the traffic information according to both the source and destination.

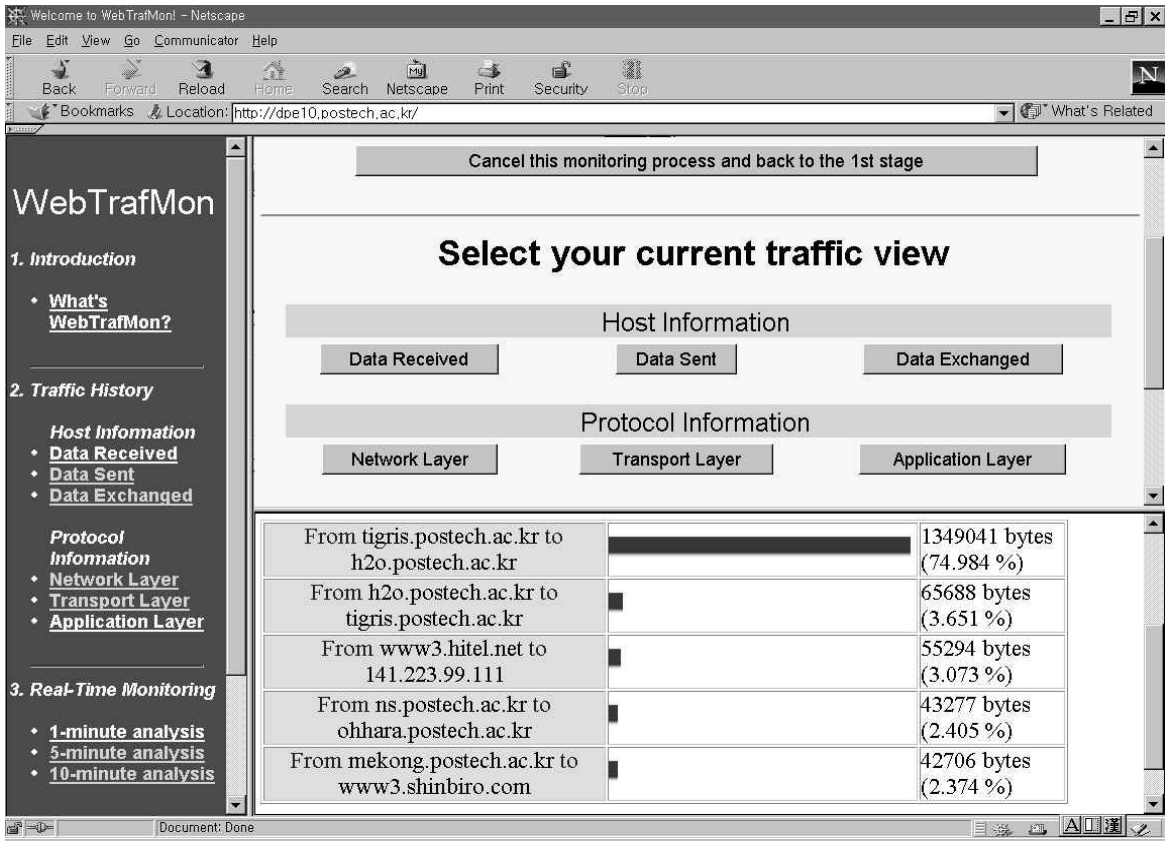

Fig. 7. Real-time traffic monitoring using WebTrafMon.

The “Protocol Information” part displays the protocol information classified according to each network layer. “Real-Time Monitoring” is used to monitor the network status in real-time, online.

Using the Web browser, users may select a monitoring time interval or review the existing traffic history. During that time period, WebTrafMon collects the network packets from a network segment and produces Web-based output results of current network status.

Fig. 6 shows sample Data Sent traffic history information. It shows both the most recent traffic status and accumulated daily traffic status. Each window can be resized by the user for better view.

Fig. 7 is the main control panel for the real time monitoring process. Users can check network statistics from this user interface by clicking on the buttons listed below:

- “Data Sent” displays which host sent the most packets.
- “Data Received” displays which host received the most packets.
- “Data Exchanged” displays the traffic between both the source and destination.
- “Network Layer” displays the traffic status about the network layers.
- “Transport Layer” displays the traffic information on the transport layer (e.g. TCP or UDP).
- “Application Layer” displays the top-level application protocols (e.g. Telnet, FTP or HTTP).

During the monitoring of the enterprise network using WebTrafMon, a big bandwidth killer, an AFS [19] file server was detected. When analyzed, the AFS file server always appeared as the dominant host that generated the most network traffic. These days, many AOD servers offer mp3 (Mpeg Layer 3) music files [20]. Listening to music through a mp3 site causes the network bandwidth to soar. If a number of users listen to music through a mp3 site online, network traffic can peak. In this situation, WebTrafMon can be used to find which server consumes the most network resources.

WebTrafMon can be configured to run automatically and periodically. This feature allowed a long-term trend and provided greater insight into the enterprise network.

7. Conclusion and future research

WebTrafMon has been developed as a Web-based network traffic monitoring and analysis system. WebTrafMon utilizes a portable network probing and packet analysis system, and provides easy-to-use, ubiquitous Web user interface. It provides network managers with real-time and

historic traffic monitoring capabilities based on destination, source and source-destination host pair as well as based on network-layer, transport-layer and application-layer protocols.

WebTrafMon can easily provide answers to questions such as “which application is consuming the most of network resources?”, “which are the top five hosts that consume the most network resources?”, and “which host pairs are generating the most network traffic?”. This ability provides more comprehensive insight into the network’s use.

Although basic features and operations were implemented for WebTrafMon, improvement remains a vast, unexplored territory. For example, we plan to analyze a larger size of log files, which the probe generates. Integrating with MRTG [1] will also be useful. Although MRTG is an excellent tool for monitoring network load, it cannot provide the kinds of host-related or protocol-related information that WebTrafMon can. Configuring MRTG to run WebTrafMon when network traffic peaks or satisfies a certain condition that the user has previously defined will be useful, providing network status details at any specific time. Another interesting and challenging future work is to extend WebTrafMon so that it can monitor and analyze switched networks (such as Fast Ethernet and Gigabit Ethernet).

References

[1] T. Oetiker, D. Rand, MRTG: Multi Router Traffic Grapher, http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html.
[2] C. Bullard, argus-1.7.beta.1b, ftp://ftp.sei.cmu.edu/pub/argus/.
[3] C. Hunt, TCP/IP Network Administration, O’Reilly & Associates, 1992.
[4] W.S., Kirchhof, Multi-protocol session analysis, http://nersp.nerdc.ufl.edu/~bill/Thesis/Thesis.html.
[5] Lawrence Berkeley National Laboratory, tcpdump 3.4a6, ftp://ftp.ee.lbl.gov.
[6] D. Perkins, E. McGinnis, Understanding SNMP MIBs, Prentice-Hall, Englewood Cliffs, NJ, 1997.
[7] A.S. Tanenbaum, Computer Networks, 2, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[8] W.R. Stevens, Unix Network Programming, 1, Prentice-Hall, Englewood Cliffs, NJ, 1998.
[9] E. Vyncke, Etherload homepage, http://www.ping.be/ethload/.
[10] T. Berners-Lee, R. Cailliau, J. Groff, B. Pollermann, World-Wide Web: the information universe, Electronic Networking 1 (2) (1992).
[11] T. Berners-Lee, R. Fielding, H. Frystyk, HyperText Transfer Protocol-HTTP/1.0, RFC 1945, IETF HTTP WG, May 1996.
[12] NCSA, The common gateway interface, http://hoohoo.ncsa.uiuc.edu/cgi/.
[13] G.R. Wright, W.R. Stevens, TCP/IP Illustrated, 2, Addison-Wesley, Reading, MA, 1994.
[14] W.R. Stevens, TCP/IP Illustrated, 1, Addison-Wesley, Reading, MA, 1994.
[15] J. Raynolds, J. Postel, Assigned numbers, RFC 1700, Network WG, October 1994.
[16] W3C, HTTP/1.1 Performance overview, http://www.w3.org/pub/WWW/Protocols/HTTP/Performance/.
[17] D. Curry, J. Mogul, nfswatch-4.3, http://ftp.lip6.fr/pub2/networking/nfs/.
[18] Apache development group, Apache web server 1.31, http://www.apache.org.
[19] Transarc Inc., AFS, http://www.transarc.com.
[20] Zco Inc., The MP3 resources on the Internet, http://www.mp3.com.