Web Server Cluster's Load Balancing for Security Session

Seok-Soo Kim, *Member, KIMICS*

Abstract – In order to create security session, security keys are preconfigured between communication objects. For this purpose, Handshake Protocol exists. The pre-master secret key that is used in this process needs to interpreted by a server to create master secret key, whose process requires a big calculation, resulting in deteriorating system's transmission performance. Therefore, it is helpful in increasing transmission speed to reuse secret keys rather than to create them at every connection.

Index Terms – Security, Session, Web, Cluster, Load balancing

I. INTRODUCTION

Current network services are implemented based on TCP/IP as the basis for standard Internet protocols [1]. To perform longest prefix match, the existing routing process handles it through software, which makes it difficult to give high-speed transmission. However, to perform high-speed transmission, forwarding engine [2, 3], enabling routing possible through hardware and making the routing process simplified, is developed, which results in high speed IP routing. However, TCP exists in transport layer and closely connects with application layer, which has a high inclination toward software. Because of this, it is not easy to apply high-speed techniques. As e-commerce transactions become more popular, demand for data service is drastically increasing via the Web. With adding varied features to the network, which carry special purposes, workloads of routers and switches comprising the network are gradually increasing. Given the fact, without considering transmission speed, the transmission performance of the overall network will naturally deteriorate.

As the interest and demand in e-commerce using web server explode, the number of network that accommodates this web server cluster architecture is increasing. Therefore, establishing a web server cluster network environment that guarantees scalability becomes an important technical issue [4, 5, 6]. Namely, when an exterior client sees a web server cluster, it is important for dispatcher to implement server management functions toward the back-end servers so that the web server cluster is seen as a single server. At this time, the dispatcher should perform an efficient session distribution policy so that back-end servers can have balanced loads among them. Meeting the tendency of high-speed network, the technology of establishing and managing high-speed TCP connections based on the technology of managing the existing multiple back-end servers in the web server cluster environment are gaining its impotence.

For this purpose, TCP Splicing has emerged as a higher speed TCP connection technique. Utilizing this technique, such technique that efficiently creates and manages sessions that are fast and have special purposes in the web sever cluster environment, will be the core technology that strengthens the current e-commerce service.

Namely, TCP Splicing is not a technique to apply to TCP/IP stack in a single system such as client and server, but a technique to connect TCP at the proxy (belongs to L4 switch) that connects client with server. In the existing proxy server, the proxy processes the headers of incoming packets as they are received from client and uploads up to application layer process that resides in the main processor in order to create connection between client and server.

Because of this reason, this study selects web server cluster as a target for transmission performance optimization. The issue of study focuses on the implementation of high-speed data transmission between client and server in such an environment of the web server cluster.

II. SECURITY SESSION

A. Security Session Strengthening Technology

In the web environment, numerous information requires security. For example, private information such as social security number and card number needs extra security. To use encryption for the information, Secure Sockets Layer (SSL) is widely used [7]. SSL is a security service that works above TCP connection. It interlinks along with application layer services such as HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transport Protocol). Thanks to the benefits of varied authentication and encryption techniques, it is recognized one of the most representative security software. Considering this influence, Internet Engineering Task Force (IETF) adopts SSL as the standard of security session service and changes its name to Transport Layer Security (TLS) and works on its standardization. However, TLS protocol does not guarantee high-speed transmission in the web server cluster environment.

In order to create security session, security keys are preconfigured between communication objects. For this purpose, Handshake Protocol exists. The pre-master secret key that is used in this process needs to interpreted by a server to create master secret key, whose process

Manuscript received April 3, 2005.

Seok-Soo Kim is with Department of Multimedia, Hannam University, Daejeon, Korea (e-mail : sskim@ hannam.ac.kr)

requires a big calculation, resulting in deteriorating system's transmission performance. Therefore, it is helpful in increasing transmission speed to reuse secret keys rather than to create them at every connection. However, increasing reuse rates of sessions to increase transmission performance is not applied to the entire network environment. In the cluster environment, excessive reuse of sessions undermines load balancing and overburdens a specific server resulting in explosive network traffics in worst case. As a result, considering a technique that balances between load balancing and security session reuse, a handshake algorithm is required to minimize the transmission speed delay in the overall networks.

B. Web Server Cluster's Load Balancing

The methods of handling client's requests are implemented in two ways: Relaying front end [4, 7] and TCP handoff [5]. First, one of the examples of relaying front-end methods is an algorithm that is suggested by Rice University. When a request happens from a client, dispatcher finds the corresponding back-end server based on the contents information like URL, distributes loads, and evaluates server's loads in real-time. When overloads happen to a specific server, loads will be distributed to other servers.

On the other hand, TCP handoff method has such architecture that the selected back-end server directly sends a reply to a client without going through the dispatcher, although dispatcher manages load distribution. The security policy that this study suggests combines the above two methods. In a specific time frame, sessions will be reused, but the session information that is stored will be regularly initialized. In this way, each server's loads need to be balanced in a long-term basis. To apply this security policy, the above methods of processing client's requests need to be implemented in advance [8].

III. LOAD BALANCING SYSTEM

A. Purpose

Due to the recent explosive increase of users connected to network and its subsequent services, user's demands cannot be satisfied only with the development of electronic technology and communications equipment. To solve this issue, load balancing that handles recent data as cell unit is suggested.

However, scheduling by cell unit requires extra works such as reassembling of data after switching. When switching the packets into a variable length not a fixed length, reassembling process is omitted in the cell unit scheduling method, which results in increasing a process rate of traffic[7,8].

B. Implementation Goal

- To provide a more stable service by traffic distribution across multiple Web servers
- A user's job can be done faster than only using a single computer by evenly distributing loads across computers. Though using tens of computers, if the work is done only in a single computer, desired performance increase is not guaranteed. That's why the load balancing is gaining its impotence.
- To evenly distribute loads, make a virtual server between client and web server. All the requests done by client will be managed by the virtual server that will decide which server will respond. At this time, client does not know which server responds.
- In order to view analyzed traffic, implements software that provides GUI environment using a window box.

C. Implementation Features : Load Balancing using Round Robin Algorithm

- Transfer protocol and application layer protocol needs to be available in order to control network traffic.
- After assuming the packets in a variable length, traffic needs to be controlled using Round Robin Algorithm.
- To visualize the analyzed traffic, it needs to be expressed in graphs.
- With software, users need to print the results in a GUI environment so that they can understand them easily .
- Implement a virtual server based on the above theory.
- The virtual server determines to which server a requesting client can be connected. The client does not know the connecting server, but the server that is connected by the virtual sever handles the request.

IV. FUNCTION MODULE

Based on the fact that Network Interface Card (NIC) receives all the packets broadcasted on the same network in addition to the packets incoming to its system, the modules will capture packets, analyze the header information of packets, output packet flow with a graph with detailed information. The modules include packet capture module, analyzing module, and output module. First, the packet capture module captures packets on the same network in real-time and sends them to the analyzing module, using Libpcap (Portable Packet Capturing Library). Second, the output module outputs the analyzed packet information including data flow (size) according to protocol with graphs and shows detailed information.

This is a program that provides library that captures packets. It is required when monitoring packets. This library consists of the followings.

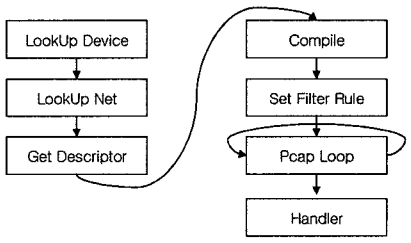

Fig. 1 Libcap Architecture

(1) Lookup Device

This is to look up a device name of NIC that is using for actual packet capture. It uses the pcap_lookupdev() function. This function returns a character string of NIC device name that is directly attached to the

network at the system. It uses as a delivery factor of pcap_openlive() and pcap_lookupnet() functions.

(2) Lookup Net

This is to find out NetID that corresponds to the device name of NIC found in the (1) process. It uses the pcap_lookup() function and its delivery factor is a character string of NIC device.

(3) Get Descriptor

This is to get descriptor for packet capture. This routine uses pcap_ open_live() function and requires five delivery factors. The first delivery factor is NIC device character string and the second is to set a size of packet to capture, which is snaplen. The reason to set a size is to consider the maximum size of packet, 1514 bytes. However, the header of packet is the first 68 bytes and the data part follows. Since the header will be enough to find the information of packet, the size is normally "68." The third is an area to designate promiscous mode, which allocates as "1." The fourth is to designate time out of packet capture. Finally, designate a buffer to save an error when an error arises.

(4) Compile and Set Filter Rule

This routine designates a rule set and libpcap follows a designated rule.

(5) Pcap Loop and Handler

The final stage of libpcap is to capture and to process the captured packet. For packet capture, this routine uses pcap_loop() function and requires four delivery factors. The first delivery factor writes packet capture descriptor obtained from the (3) stage. Secondly, the number of packet to capture is designated. When the value is inputted as "-1", it loops limitlessly and captures packet. Thirdly, handler function's pointer is designated and pcap_loop() function calls designated handler function to process whenever it captures packet. The last factor is a user-defined factor and it is generally designated as "null."

(6) Network device and packet capture that exist in the server

It detects network adapter in the server and finds out IP address and subnet mask. It detects packet that comes into this device. At this time, specific device and specific protocol can be configured.

ACKNOWLEDGMENT

This work was supported by a grant No. (R12-2003-004-03003-0) from Ministry of Commerce, Industry and Energy.

V. CONCLUSION

The implemented package software measured packet volume that was generated from data generator, virtual server, and server 1, 2, 3, and could find out traffic distribution toward Server 1, 2, 3.

As the result of the study shows, Round Robin Algorithm ensured definite traffic distribution, unless incoming data loads differ much. Although error levels were high in some partial cases, they were eventually alleviated by repeated tests for a longer time.

In order to create security session, security keys are preconfigured between communication objects. For this purpose, Handshake Protocol exists. The pre-master secret key that is used in this process needs to interpreted by a server to create master secret key, whose process requires a big calculation, resulting in deteriorating system's transmission performance. Therefore, it is helpful in increasing transmission speed to reuse secret keys rather than to create them at every connection.

REFERENCES

[1] PACS pages : Eric John Finegan's PACS / Telemedicine Resourcepage,http://www.dejarnette.com/dfinegan/pacspage.htm.

[2] T. Gotwald, M. Daniaux, A. Stoeger, R. Knapp, and D. Nedden, "The value of the World Wide Web for teleeducation in radiology," Journal of Telemedicine and Telecare, Vol. 6 No. 1, May 2000.

[3] M. Sohlenkamp and G. Chwelos, "Integrating Communication, Cooperation, and Awareness : The DIVA Virtual Office Environment," Proceedings of the ACM Multi- media '92, pp. 331- 343, April 1992.

[4] A Lange, J-P Q V van de Ven, B A L Schrieken, B Bredeweg and P M G Emmelkamp, Internetmediated, protocol-driven treatment of psychological dysfunction., Journal of Telemedicine and Telecare, Vol. 6, No. 1, 2000.

[5] Hyun Cheol Jeong, "Multilevel Secure Recovery Management of Medical Databases in Hospital Information System," Journal of Korean Society of Medical Informatics, Vol. 6, No. 2, pp. 17-25, June 2000.

[6] Kilgore C., "Patients take the wheel with internet health records," Telehealth Magazine, Vol. 5, No. 7, Dec. 2000.

[7] Seok Soo Kim and Dae Joon Hwang, "Telemedicine Multimedia Database on the Cyber Doctor," VIProm Com-2001, Zadar, Croatia, June 2001.

[8] Seok Soo Kim and Dae Joon Hwang, "An Algorithm for Formation and Confirmation of Password for Paid members on the Internet-based Telemedicine," Springer, LNCS 2105, pp. 334-340, June 2001.

Seok-Soo Kim

Received a B.S. degree in computer engineering from Kyungnam University 1989, and M.S. degree in Information engineering from Sung kyun-kwan University 1991 and Ph D. degree in Information engineering from Sung kyunk-wan University 2002.

In 2003 he joined the faculty of Hannam University where he is currently a professor in Department of Computer & Multimedia Engineering. His research interests include Multimedia Communication systems, Distance learning, Multimedia Authoring, Telemedicine, Multimedia Programming, Computer Networking. Information Security. He is a Member of KICS, KIMICS, KIPS, KMS, and DCS.