IEICE TRANS. COMMUN., VOL.E88–B, NO.5 MAY 2005

INVITED PAPER *Joint Special Section on Recent Progress in Optoelectronics and Communications*

A 10 Gb/s Firewall System for Network Security in Photonic Era

Masaru KATAYAMA[†a)], **Hidenori KAI**[†], **Junichi YOSHIDA**[†], ***Members***, **Masaaki INAMI**[†], ***Nonmember***, **Hiroki YAMADA**[†], **Kohei SHIOMOTO**[†], ***and*** **Naoaki YAMANAKA**[†*], ***Members***

SUMMARY Although the Internet is playing an increasingly significant role in global communication, it remains vulnerable to malicious traffic such as worms and DoS/DDoS attacks. In the last few years, the emergence of high speed active worms, such as Code Red II, Nimda, SQL Slammer and MS Blaster, has become a serious issue. These worms cause serious damage to communication networks throughout the world by using up network bandwidth. In addition, since conventional firewall systems are located just in front of the server and do not prevent malicious traffic from entering the network, they cannot prevent such network congestion. Therefore, the firewall between domains or between core routers should play important roles in the photonic networks. We have developed a prototype system of a network firewall using reconfigurable processors. In this paper, we overview the developed system and present its evaluation results.

key words: *DDoS, worm*

1. Introduction

Since the emergence of the Internet, malicious activity by worms, scan, Denial of Service (DoS) attack and distributed DoS (DDoS) attack tools has been a constant problem. Since the first Morris worm [1] was released over ten years ago, Internet worms have continued to emerge. Although the Internet is faster and more convenient to use than before, it remains vulnerable. In fact, Code Red II [2] and Nimda [3] in 2001, SQL Slammer [4] and Blaster [5] in 2003, and MyDoom [6] and NetSky [7] worms in 2004 caused serious damage throughout the world. In the high-speed photonic network, in particular, malicious Internet traffic threatens to congest network bandwidth. Worms are easy to create and many user-friendly attack tools are widely available today.

The appearance of the Code Red II worm accelerated the analysis of high-speed active worms modeling [8], [9]. However, no effective prevention method has been proposed. Several methods have been proposed to prevent DoS/DDoS attacks [10]. The traceback techniques [11], [12] for DoS/DDoS prevention are effective. However, the method takes a long time to detect the attackers, and it cannot detect random-spreading attacks such as worms. For this reason, a system of preventing worms from infecting networks in the first place is necessary.

We propose a distributed firewall system to counter the worms. This firewall architecture consists of firewalls located at the border of the network, and orchestrates them to prevent malicious traffic from entering the network. In order to locate the border of the network, the firewall system requires the following features:

- 10 Gb/s packet processing performance and wire-speed traffic handling performance,
- attack detection for undefined victims and the prevention of malicious traffic,
- a hitless firewall algorithm update function without interrupting services.

We have developed a firewall system with these features. The rest of this paper is organized as follows: Sect. 2 overviews well-known DDoS attacks and worms. Section 3 describes our distributed firewall architecture. Section 4 describes the required features of our firewall system. Section 5 discusses its implementation using the reconfigurable processors and presents the evaluation results. Finally, Sect. 6 concludes the paper.

2. DDoS Attack and Worm

In this section, we introduce two well-known attacks. One is "SYN flooding" and the other is the "SQL Slammer" worm.

2.1 SYN Flooding

"SYN flooding" is the most well-known attack. This attack exploits the design weakness of TCP's three-way handshake mechanism. In the TCP's three-way hand-shake mechanism, the client sends the SYN packet to the server. The server sends back the SYN-ACK packet to the client, leaving the session in "half-open" state. Finally the client sends back the ACK packet to the server, and the server changes the state of the session to "open" state. During the half-open state, the server waits for the ACK packet to be sent back to the client. SYN flooding occurs when a large number of TCP SYN packets is sent to a victim's port. If the client intentionally does not send the ACK packet, the server waits 75 seconds for the timer to expire. The malicious user sends a huge amount of SYN packets instantaneously to the victim server so that the server resources for the half-open session state are consumed and new TCP session establishment attempts are rejected.

Manuscript received January 11, 2005.

[†]The authors are with the Network Service Systems Laboratories, NTT Corporation, Musashino-shi, 180–8585 Japan.

[*]Presently, with the Department of Science and Technology, Keio University.

a) E-mail: katayama.masaru@lab.ntt.co.jp

DOI: 10.1093/ietcom/e88–b.5.1914

2.2 SQL Slammer Worm

This worm targets SQL Server computers and is self-propagating malicious code that exploits the vulnerability described in VU#484891 (CAN-2002-0649) [13]. This vulnerability allows for the execution of arbitrary code on the SQL Server computer after stack buffer overflow. This worm attacked the global Internet on January 24, 2003 and is the fastest computer worm yet seen. The Slammer worm doubled in number every 8.5 seconds during the explosive first minute of its attack. Within 10 minutes of birth, 5:30 a.m. (UTC) Jan. 25 (9:30 p.m. PST, Jan. 24), the worm was observed to have infected more than 75,000 vulnerable hosts. Thousands of other hosts may also have been infected worldwide. The infectious hosts spread billions of copies of the worm into cyberspace, significantly slowing Internet traffic, and interfering with many business services that relied on the Internet.

Although this particular computer worm did not carry a malicious payload, it caused a lot of harm by spreading so aggressively and congesting networks.

The Slammer's tiny size, 376 bytes, enabled it to reproduce rapidly. In comparison, the Code Red worm spread much more slowly not only because it took longer to replicate, but also because infected machines sent a different type of message to potential victims that required them to wait for responses before subsequently attacking other vulnerable machines.

The Code Red worm ended up infecting 359,000 hosts, in contrast to the approximately 75,000 machines that the Slammer hit.

A computer infected with the Slammer worm and provided with a one-megabit-per-second connection was capable of sending out 300 copies of Slammer each second. A single computer with a 100-megabit-per-second connection, like those found at many universities and large corporations, would allow the worm to scan 30,000 machines per second. The novel feature of this worm, compared to all the other worms that we have studied, is its incredible speed.

3. Distributed Firewall System

3.1 Distributed Firewall Architecture

The proposed distributed firewall architecture consists of firewalls located at the border of network, and orchestrates them to prevent malicious traffic from entering the network. Conventional firewall systems are located just in front of the server and do not prevent malicious traffic from entering the network and thus cannot prevent network congestion. An overview of the architecture is shown in Fig. 1. The architecture has two features. One is the protection of malicious traffic as DDoS attacks and worms on the border routers in order to prevent traffic congestion. The other is that the border router advertises the attack information and its filtering rules when the router detects the attacks. The advantage of

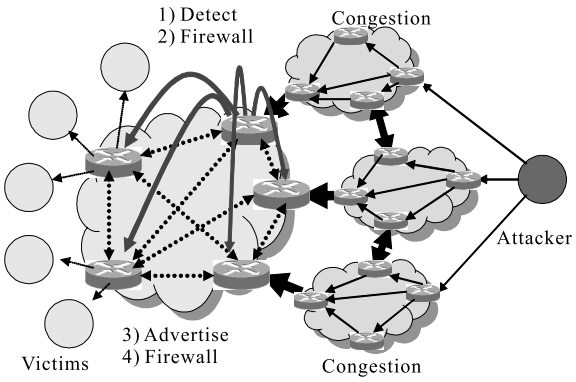


Fig. 1 Distributed firewall architecture.

the proposed architecture is that the inside of the network is enclosed by border routers that protect from the attacks. The proposed method consists of four steps:

1) attack detection,
2) filter creation on the detected router,
3) attack information advertising,
4) filter setting on the information-received border router.

1) Attack detection

This part is generally called IDS (Intrusion Detection Systems). In the proposed architecture, the firewalls are set on the border routers between the victim and the attacker. Therefore, attacks that are detected at border router are blocked. To classify the packets, layer 2–4 information is used.

2) Filter creation on the detected router

The attack detection is performed by the firewall on the detected firewall. The rule is immediately created after the attack is detected. The malicious packets are discarded on the border firewall. However, the bandwidth resource is still consumed by malicious packets in the network because they are allowed to enter the network via other routes.

3) Attack information advertising

The attack information is exchanged between the border firewalls. Our key technology is to use the BGP (Border Gateway Protocol) [14] extensions because our target network is not the Internet but the managed network. We add new attributes for advertising the filters for attacks and attack information such as detected attack type, attack time and so on. In this way, all border firewalls take countermeasures to prevent the malicious packets from entering the network.

4) Filter setting on the information-received router

If the border firewall receives the attack information, a new rule is added to the filtering table, and the attacks are orchestrally blocked by border firewalls. In this way, the malicious packets are prevented from consuming network resources.

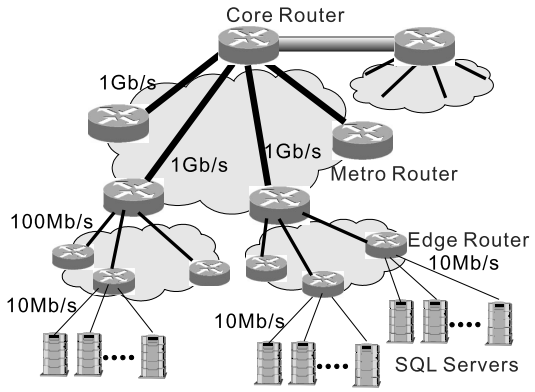


Fig. 2 Network model.

3.2 Performance Evaluation of the Distributed Firewall System

In order to evaluate the distributed firewall against SQL Slammer worm, we assumed the network which consists core routers, metro routers, edge routers and SQL servers as shown in Fig. 2. In Fig. 2, the core routers are linked to each other with photonic links. The 16 metro routers are linked to a core router with 1 Gb/s. The 32 edge routers are linked to a metro router with 100 Mb/s. The 256 SQL servers are linked to an edge router with 10 Mb/s. The system in our simulation consists of $N = 256 * 32 * 16 = 1301072$ servers.

In our simulation, a host is in one of three states at any time: susceptible, infected, or removed. At the beginning of the simulation, a randomly selected host is initially infected and the others are all susceptible. The infected host sends out multiple copies of the worm to other hosts, whose address is randomly generated, at full line speed, and the size of Slammer worm packet is 376 bytes. The infection delay time is defined as the time for the worm to travel from the infected host to the second one. In our simulation, we assume the infection delay time of 0.1 sec. An infected host will not change its infection behavior if it is infected again by other copies of the worm. To capture the impact of cleaning, patching and filtering on worm propagation, we assume that the number of immunized hosts increases with time. At each discrete time, we randomly choose some non-immunized hosts for immunization regardless of whether they are infected or still susceptible.

We investigated the effect of the proposed distributed firewall system. We compare it with the effect of a system where there are no firewalls located at the border. In our numerical analysis, we evaluate residual traffic.

Figure 3 shows the residual bandwidth as a function of the elapsed time. The residual bandwidth of the conventional system (dashed-line) falls within the first 400 seconds. This is due to the fact that the number of infected hosts is increasing and residual bandwidth is consumed by the worm traffic. After 400 seconds, it allows the residual bandwidth to recover, because the worm traffic is being throttled which suppresses the rate of host infection. The continuous-line

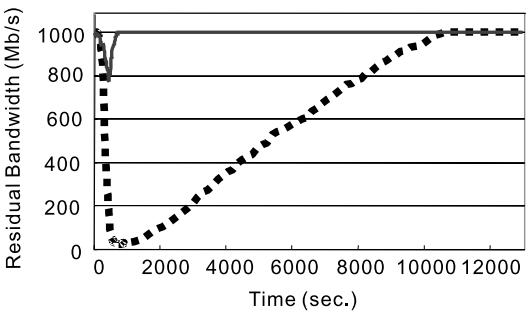


Fig. 3 Residual bandwidth.

shows the residual bandwidth of the proposed distributed firewalls. The figure indicates that the distributed firewalls can keep almost all of the bandwidth clear of worm traffic. The results indicate that the proposed distributed firewall system is more effective than the conventional system in blocking the SQL Slammer worm.

4. Firewall Systems Located on the Border

The proposed distributed firewall system prevents the malicious traffic entering the network in order to locate on the border of the network. The features of the proposed firewall system are different from the features of conventional firewall systems. Each firewall system has the following features.

Conventional firewall system

- The firewall is located in front of the servers and on the gateway of LAN.
- The firewall guards defined victims.
- All packets both upstream and downstream pass through the firewall.
- The firewall system takes maintenance time in order to update its functions.

Distributed firewall system located on the border

- The firewall is located on the border of the network.
- The firewall guards the network resources, and the victims are not defined.
- All packets with one session do not have to pass through the same borders.
- The firewall handles more than 10 Gb/s traffic.
- The firewall system has a hitless update function without interrupting services.

In particular, a firewall system located on the gateway between ISPs (Internet Service Providers) must have these features, and we are not able to introduce the conventional firewall systems.

4.1 Attack Detection without Defined Victims

The conventional firewall system defines the destination address (DA) of the victim servers and network address. Then,

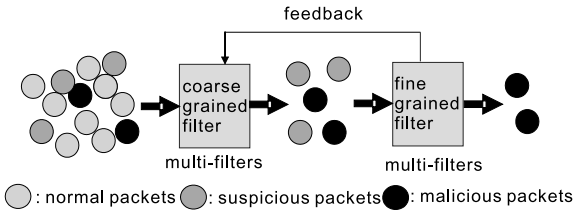
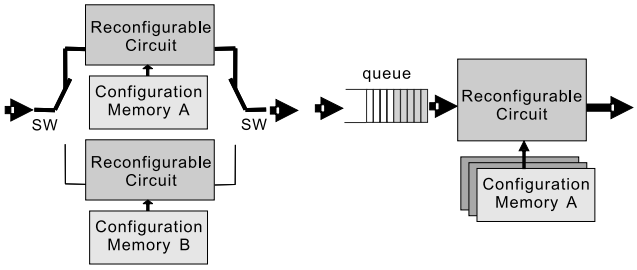


Fig. 4 How to detect the malicious traffic.



(a) Switch-type (b) Buffer-type

Fig. 5 How to update new circuit.

packets with the undefined DA are discarded. However, a firewall system located on the gateway does not define the victims. There are a huge number of destination addresses in the network in addition to the transit traffic. We propose a technique of detecting the malicious traffic among many flows. An overview of the technique is shown in Fig. 4. The traffic in the network consists of normal packets and malicious packets. Some normal packets behave like malicious packets when they down on the wire. These packets are suspicious packets. We use multi filters with different functions in order to detect the malicious packets step by step.

Below we present two examples to show how the multi filters work, one is for DDoS attacks and the other is for worms. In the case of DDoS attacks, many attackers attack a victim server. The malicious packets have the same destination address. Traffic volume with the same DA increases suddenly through the gateway. If we can detect traffic abnormality with the same DA, we can use the DDoS detection algorithms [10].

In the case of worms, the infected server sends out multiple copies of the worm to other hosts at full line speed. The destination address of the worm is randomly generated and/or the network address which the infected server belongs to such as 192.168.10.0/24. The second infected server sends out multiple copies of the worm. Thus, the worms are rapidly spread and have various DA. However, the worms have the same port number and the same protocol as the vulnerability of the OS. For example, SQL Slammer has UDP 1434 port, and Blaster has TCP 135 port. We can detect the traffic abnormality with the same keyword.

4.2 Hitless Update Function

Up to now, high-speed packet processing functions have always been implemented with ASICs so the circuitry was hard wired in and unmodifiable. In order to speed up or upgrade packet processing, the only alternative was to pull out the old board and replace it with a new board, and this created enormous problems in terms of maintenance and operability. When flexibility to modify functions was a priority and network processors were used, the processing was performed by software, which limits the speed to no more than about several hundred megabits per second.

The firewall system must be updated to the new filter rule and/or the new algorithm whenever a new attack is detected. The conventional firewall systems usually require maintenance time in order to update the functions and thus interrupt services. However, a firewall system located on the gateway between ISPs must have a hitless update function without interrupting services. In addition, the firewall system handles more than 10 Gb/s traffic. In order to combine flexibility with high performance, we use reconfigurable devices. We can exchange the configurations by rewriting the configuration memories.

There are two type of hitless update function using reconfigurable devices [15]. One is a switch type implementation, the other is a buffer-type implementation. These implementations are shown in Fig. 5.

The switch-type implementation exchanges the configurations by using switching devices. One reconfigurable device has a main configuration memory, and the other device has a new configuration memory. The problem with this type implementation is that it requires twice the volume of hardware devices. On the other hand, the buffer-type implementation uses dynamic reconfiguration devices [16]. This type of device has several reconfiguration memories. One reconfiguration memory is written as the main configuration, and the other reconfiguration memory is written as a new configuration. By dynamically exchanging the reconfiguration memories, we can realize new functions. The buffer-type, however, has the buffer storing the packets during the reconfiguration. A feature of this type requires less hardware volume than the switch-type. In addition, higher speed devices can be developed for the hardware, reducing the exchange time.

5. Implementation and Evaluation Results

In order to develop a firewall system located on the gateway, we selected the reconfigurable device, DAPDNA-2 [17]. The DAPDNA-2 is a reconfigurable processor with 6 channels of 5 Gb/s interface. This processor can handle more than 10 Gb/s traffic with 166 MHz processing. In addition, it has four dynamic reconfiguration memories, and exchanges within one clock (about 6 nsec.). The buffer-type implementation is easy, and requires only a small buffer. Moreover, detection without undefined victims is easily implemented using the distributed 32-bits ALUs and the distributed memories on the device as shown in Fig. 6.

Figure 7 is the photograph of the developed prototype system, and Fig. 8 shows an overview of the implementation. This system has 4-port 2.5 Gb/s POS interfaces, and 4-

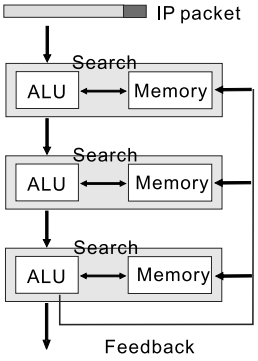


Fig. 6 An overview of implementation using distributed ALUs and memories.

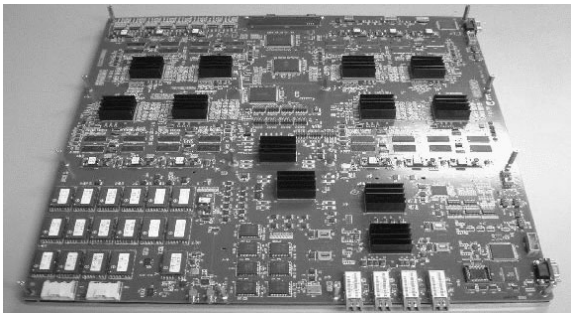


Fig. 7 Prototype system.

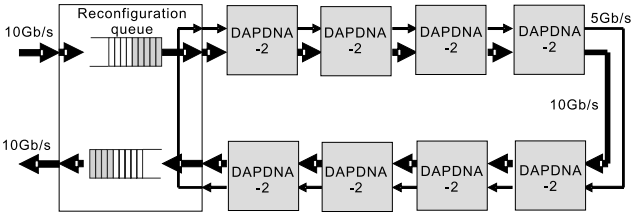


Fig. 8 An overview of the implementation.

port data is multiplexed. It is tandem implemented on eight DAPDNA-2 chips with 10 Gb/s data and 5 Gb/s control data. In order to hitless update, the buffer is implemented on the system. The system with Layer 4 flow processing and a hitless update function is called WSPEED (Wire-Speed Packet Engine for EDge system).

The firewall system located on the gateway can be realized on the prototype system installing the firewall configuration. The system has the following features:

- 10 Gb/s Layer 4 traffic handling,

  The system supports the L4 flows with network address.

- protecting the network resources even if the victims are not defined,
- hitless update function.

Figure 9 shows the evaluation network used to verify the hitless update function and the SQL Slammer worm protection. The evaluation results of the hitless update function

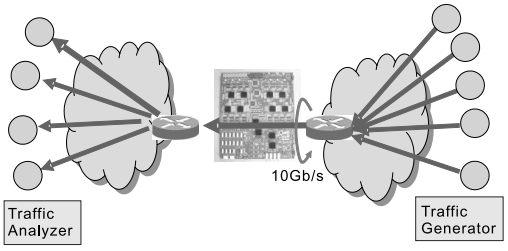


Fig. 9 Evaluation network I.

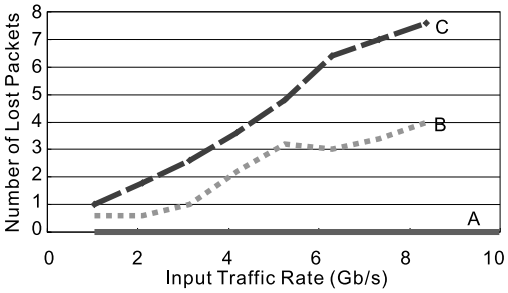


Fig. 10 Number of lost packets.

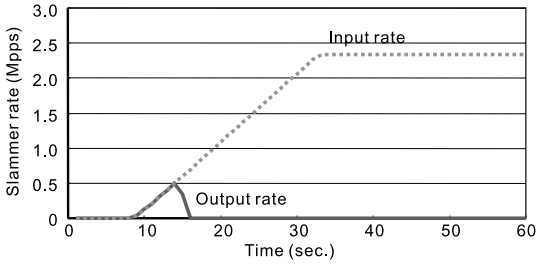


Fig. 11 The traffic rate of Slammer worm.

is shown in Fig. 10. The line A is the number of lost packets to reconfigure function using the proposed hitless update, and both line B and C are them without the queues. In the case of B, a middle-size function is updated to a small-size function, and a large-size function is updated to a small-size function in the case of C. The line A shows that no packet is lost when a function is updated, and it is independent to function size. Therefore, we can confirm the hitless update function.

In order to evaluate the efficiency against SQL Slammer worm, the traffic generator sends Slammer packets to the traffic analyzer. The generator sends to a randomly selected destination addresses. Figure 11 shows the input rate of Slammer worm (dashed-line) and its output rate (continuous-line) as a function of the elapsed time. This graph shows that our system detects the Slammer worm increasing the traffic rapidly. Therefore, our system protects against Slammer worm.

Figure 12 shows the another evaluation network used to verify our firewall system. The system transfers 7.5 Gb/s (2.5 Gb/s $\times$ 3) background traffic and 2.5 Gb/s traffic including MPEG streaming and DDoS attack traffic. The attacker

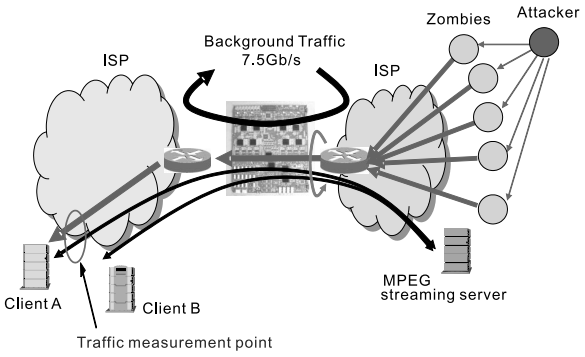


Fig. 12 Evaluation network II.

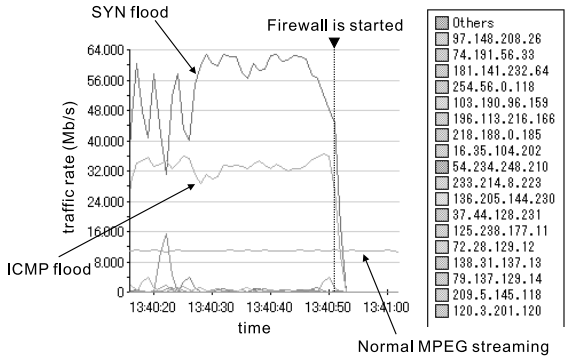


Fig. 13 DDoS traffic at the measurement point.

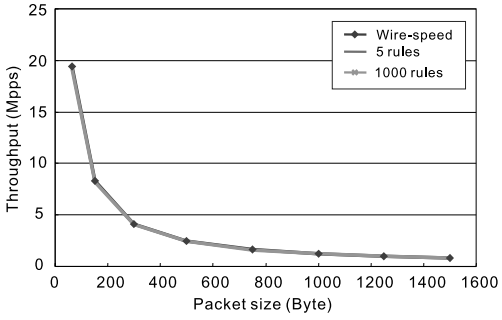


Fig. 14 Relationship between packet size and throughput.

sends the spoofed IP address packets to the MPEG client A.

Figure 13 shows a snapshot of the experimental results. There is a relationship between time and the flow rate. The flat-line is MPEG streaming flow. The dashed-line time is hitless reconfiguration time. Before the dashed-line time, ICMP flooding and SYN flooding with the spoofed addresses are transferred to Client A. In this time, the configuration is the through function. At the firewall-started time, we have reconfigured the firewall system. Our system does not affect anything for the normal MPEG flow of Client A. After the dashed-line, our system protects against ICMP flooding and SYN flooding. On the other hand, no packets are discarded in the normal MPEG flow of Client B and the background traffic.

Figure 14 shows the relationship between packet size and throughput. Three lines are overlapped. Therefore, our system does not depend on the packet size or the number of firewall rules.

6. Conclusion

We developed a prototype system of the firewall system located on the gateway between ISPs using the reconfigurable processors. Its features are hitless update function, 10 Gb/s wire speed packet processing, and detection of the malicious traffic among 10 Gb/s L4 flows. Experimental results confirmed that the developed firewall system allows wire-speed packet processing and prevents the malicious traffic.

References

[1] E.H. Spafford, "The internet worm incident," ESEC'89 2nd European Software Engineering Conference, pp.446–468, Coventry, United Kingdom, 1989.
[2] CERT. CERT Incident Note IN-2001-09 "Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL, http://www.cert.org/incident_notes/IN-2001-09.html
[3] CERT. CERT Advisory CA-2001-26 Nimda Worm, http://www.cert.org/advisories/CA-2001-26.html
[4] CERT. CERT Advisory CA-2003-04 MS-SQL Server Worm, http://www.cert.org/advisories/CA-2003-04.html
[5] CERT. CERT Advisory CA-2003-20 W32/Blaster worm, http://www.cert.org/advisories/CA-2003-20.html
[6] CERT. CERT Incident Note IN-2004-01, http://www.cert.org/incident_notes/IN-2004-01.html
[7] CERT. CERT Incident Note IN-2004-02, http://www.cert.org/incident_notes/IN-2004-02.html
[8] Z. Chen, L. Gao, and K. Kwiat, "Modeling the spread of active worms," Proc. IEEE Infocom, vol.3, pp.1890–1900, 2003.
[9] M. Garetto, W. Gong, and D. Towsley, "Modeling malware spreading dynamics," Proc. IEEE Infocom, vol.3, pp.1869–1879, 2003.
[10] R.K.C. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," IEEE Commun. Mag., vol.40, no.10, pp.42–51, Oct. 2002.
[11] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source," Proc. 14th System Administration Conference, pp.319–327, New Orleans, USA, Dec. 2000.
[12] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S.T. Kent, and W.T. Strayer, "Single-Packet IP traceback," IEEE/ACM Trans. Netw., vol.10, no.6, pp.721–734, Dec. 2002.
[13] Vulnerability Note VU#484891, "Microsoft SQL server 2000 contains stack buffer overflow in SQL server resolution service," http://www.kb.cert.org/vuls/id/484891
[14] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," RFC1771, March 1995.
[15] H. Yamada, H. Kai, and J. Yoshida, "Using re-configurable technology in high-speed packet processing," Proc. ICN'04, pp.628–634, Feb. 2004.
[16] S. Trimberger, D. Carberry, A. Johnson, and J. Wong, "A time-multiplexed FPGA," Proc. IEEE Symposium on FPGAs for Custom Computing Machines, pp.22–29, April 1997.
[17] IPFlex Inc., http://www.ipflex.com/

Masaru Katayama is a senior research engineer, NTT Network Service Systems Laboratories at Musashino, Tokyo, Japan. He received the B.E. and M.E. degrees from Hokkaido University in 1990 and 1992, respectively. He has been engaged in research on system LSI design and its design methodologies using rapid prototyping systems. He has accumulated considerable experience in development of a telecommunication-oriented FPGA, called "PROTEUS-Lite", and its developing software systems since joining Nippon Telegraph and Telephone Corporation (NTT) Laboratories in 1992. His current research interests are in a high-performance IP packet processing system, its control system with Field Programmable hardware systems (such as FPGAs and Reconfigurable Processors).

Hidenori Kai received the B.E. degree in mechanical engineering from Ibaraki University, Hitachi, Japan, in 1983, In 1983 he joined Nippon Telephone and Telegraph Public Corporation and he was engaged in research and development of ATM switching prototype systems. Since joining NTT Network Service Systems Laboratories in 1996, he has been engaged in research and development of frame-relay node systems, packet node systems, and reconfigurable processor systems. He is currently a Research Engineer belonging to NTT Network Service Systems Laboratories, NTT, Musashino, Japan.

Junichi Yoshida received B.E.,M.E. degree in Department of Applied Chemistry, Faculty of Engineering from Ehime University in 1994,1996. He joined Nippon Telephone and Telegraph Corporation in 1996. He had been engaged in NTT Network Service Systems Laboratories, NTT, Tokyo, Japan, since 1998, where he research and development of ATM switching systems, frame-relay node systems, packet node systems, and reconfigurable systems.

Masaaki Inami received the B.E. degree from Kanazawa University in 2002 and the M.E. degree from Japan Advanced Institute of Science and Technology in 2004. In 2004, he joined NTT Network Service Systems Laboratories at Musashino, Tokyo, Japan. He has been engaged in research on infrastructure of home networks using Bluetooth network and its design methodologies. His current research interests are in high-performance traffic analysis systems, and high-availability packet forwarding systems.

Hiroki Yamada received the B.S. degree in electronic engineering from the Tokyo Metropolitan University, Tokyo, Japan, in 1980. In 1980 he joined Nippon Telephone and Telegraph Public Corporation, where he was engaged in research and development of subscriber line interface circuits for digital telephone switching systems, a packet communication controller for packet switching systems, and ATM switching systems. He is a Senior Research Engineer, Supervisor, NTT Network Service Systems Laboratories, Tokyo, Japan. He is currently engaged in research and development of IP services switch.

Kohei Shiomoto is a Senior Research Engineer, Supervisor, at NTT Network Service Systems Laboratories, Japan. He joined the Nippon Telegraph and Telephone Corporation (NTT), Tokyo, Japan in April 1989, where he was engaged in research and development of ATM traffic control and ATM switching system architecture design. From August 1996 to September 1997, he was engaged in research on high-speed networking as a Visiting Scholar at Washington University in St. Louis, MO, USA. From September 1997 to June 2001, he was directing architecture design for high-speed IP/MPLS label switch router research project at NTT Network Service Systems Laboratories, Tokyo, Japan. Since July 2001 he has been engaged in the research fields of photonic IP router design, routing algorithm, and GMPLS routing and signaling standardization at NTT Network Innovation Laboratories. He received the B.E., M.E., and Ph.D. from Osaka University, Japan, in 1987, 1989, and 1998. He is a member of IEICE, IEEE, and ACM. He is a Secretary of International Affair of the Communications Society of IEICE. He is a Vice Chair of Information Service of IEEE ComSoc Asia Pacific Board. He was engaged in organization of several international conferences including HPSR 2002, WTC 2002, HPSR 2004, and WTC 2004. He received the Young Engineer Award from the IEICE in 1995.

Naoaki Yamanaka graduated from Keio University, Japan where he received B.E., M.E. and Ph.D. degrees in engineering in 1981, 1983 and 1991, respectively. In 1983 he joined Nippon Telegraph and Telephone Corporation's (NTT's) Communication Switching Laboratories, Tokyo Japan, where he was engaged in research and development of a high-speed switching system and high-speed switching technologies for Broadband ISDN services. Since 1994, he has been active in the development of ATM base backbone network and system including Tb/s electrical/optical backbone switching as NTT's Distinguished Technical Member. He is now researching future optical IP network, and optical MPLS router system. He is currently professor of Department of Information and Computer Science, Faculty of Science and Technology, Keio University. He has published over 112 peer-reviewed journal and transaction articles, written 82 international conference papers, and been awarded 174 patents including 17 international patents. Dr. Yamanaka received Best of Conference Awards from the 40th, 44th, and 48th IEEE Electronic Components and Technology Conference in 1990, 1994 and 1998, TELECOM System Technology Prize from the Telecommunications Advancement Foundation in 1994, IEEE CPMT Transactions Part B: Best Transactions Paper Award in 1996 and IEICE Transaction Paper award in 1999. Dr. Yamanaka is Technical Editor of IEEE Communication Magazine, Broadband Network Area Editor of IEEE Communication Surveys, Former Editor of IEICE Transaction, TAC Chair of Asia Pacific Board at IEEE Communications Society as well as Board member of IEEE CPMT Society. Dr. Yamanaka is an IEEE Fellow.