FPGA Based TCP Session Features Extraction Utilizing Off-Chip Memories

Satoshi Fuchigami

Graduate School of Information Science
Nagoya University
Nagoya,464-8601 Japan
Email: fuchigami@net.itc.nagoya-u.ac.jp

Hajime Shimada , Yukiko Yamaguchi

Information Technology Center
Nagoya University
Nagoya,464-8601 Japan
Email: {shimada, yamaguchi}@itc.nagoya-u.ac.jp

Hiroki Takakura
National Institute of Informatics
Tokyo,101-8430 Japan
Email: takakura@nii.ac.jp

***Abstract*—In recent years, unknown attacks, such as zero-day attacks and targeted attacks, have been increasing. These attacks are difficult to detect because the information gathered from already known attacks is not useful for their detection. An anomaly-based Network Intrusion Detection System(IDS) has the potential to find these attacks. However, almost all anomaly-based Network IDSs are implemented as software, so they cannot catch up with the growing network traffic. To alleviate this problem, there is Hardware/Software(HW/SW) cooperated Network IDS which migrates Transmission Control Protocol(TCP) feature extraction process to Field Programmable Gate Array(FPGA). However, the prior implementation is completed in FPGA, so that it cannot treat long TCP sessions because of shortage of memory blocks in FPGA-chip. In this paper, we propose TCP session feature extraction and cumulation by FPGA combining off-chip Ternary Content Addressable Memory(TCAM) and Dynamic Random Access Memory(DRAM) for HW/SW cooperated Network IDS. This approach uses these off-chip memories for buffering features while a TCP session continues. We present here the architecture design and implementation. We estimate that our system can manage 1,024K sessions simultaneously.**

Keywords–Anomaly Based Network IDS; FPGA; TCP Session Feature Extraction

I. INTRODUCTION

In recent years, cyber attacks have increased and more sophisticated, so that it is important to detect their invasion by monitoring network traffic. However, the amount of network traffic is growing rapidly and it requires more throughput to Network IDS. Furthermore, to alleviate these attacks, inspection of the internal network is also effective but it requires ten times larger throughput compared to Wide Area Network(WAN) gateway based inspection.

To resolve this problem, there are several studies using FPGA which is a re-programmable hardware for Network IDS. But past FPGA based Network IDS is only done in signature-based Network IDS. On the other hand, we have performed a study about HW(FPGA)/SW cooperated Network IDS [1] [2], which is implemented based on anomaly-based scheme and suited to detect increasing unknown attacks. We use FPGA for extracting TCP session features as a part of the system. It reduces the burden on the Network IDS software by migrating the feature extraction process to FPGA which occupies around 90% of CPU time [2]. However, in a previous implementation, the FPGA could not handle long and a large number of sessions because the implementation utilizes Random Access Memory(RAM) in the FPGA-chip whose capacity is quite small, i.e., 5.675Mbytes.

This paper describes a TCP session feature extraction system, which utilizes both off-chip TCAM and DRAM. The proposed system assists the Network IDS which uses PAYL [3] algorithm by implementing heavy feature extraction tasks into FPGA. When the system starts TCP session feature cumulation by a SYN packet, the proposed system prepares entry for buffering feature into both memories. Until TCP session finishes, the proposed system extracts features from TCP packets of the same session one by one and cumulates TCP session features using prepared entries. When the TCP session finishes, the proposed system outputs the TCP session feature to software side which is executed in general-purpose server machine.

The rest of this paper is organized as follows. Section II describes two tyeps of Network IDSs and a research about using FPGA for Network IDS. Section III addresses details of our proposal. Section IV explains the implementation. In section V, we estimate throughput for traffic feature extraction. Section VI concludes this research and suggest approaches for our future study.

II. RELATED WORK

There are two types of Network IDSs: signature-based Network IDS and anomaly-based Network IDS. The former detects attacks by comparing traffic data with signatures made from patterns of known attacks. This kind of method works well against known attacks but not against unknown attacks increasing today. Currently, these kinds of methods are widely used in the world and Snort [4] is one of the most famous software implementations.

The latter identifies attacks by statistically analyzing traffic features like clustering method such as K-means [5] and

 ISBN: 978-1-61208-435-0

One-Class Support Vector Machine(SVM) [6]. Those types of Network IDSs have the potential to detect unknown attacks, so that it is supposed to catch up to latest cyber attacks.

However, the current network traffic is already significant and continuously increasing. Network IDS is required to catch up with traffic in this environments. There are researches about implementing Network IDS using FPGA or Application Specific Integrated Circuit(ASIC) to alleviate this problem. Katashita et al. [7] proposed a 10Gbps throughput signature-based Network IDS using FPGA. This system inspects traffic data by traffic data signature matching method which is categorized into signature-based method. They also developed a tool which generates a circuit from Snort rules. On the other hand, hardware implementation of anomaly-based Network IDS is not generic because their detection algorithms are often difficult to implement in hardware.

III. Details of Our Proposed System

A. Concepts of the System

Future network traffic is expected to be subjected to many unknown attacks under a huge amount of traffic. To confront this situation, our proposing system aims to achieve high throughput and anomaly intrusion detection. As shown in Figure 1, in the existing anomaly intrusion detection method, traffic data are mirrored on switch and their copy are temporarily stored into storage. Then, intrusion detection process analyzes the stored data later. On the other hand, our proposed system aims at real time processing by reducing the burden of the server performing the analysis by extracting network traffic features on FPGA. This is a kind of HW/SW supported IDS. The FPGA also includes L2 switch functions, so that network traffic features extraction is done with port based distributed processing.

B. Baseline and Functions

We use Altera Stratix V GX (model: 5SGXEA7H2F35) FPGA. This board also has 20 Small Form factor Pluggable+(SFP+) ports and we already implemented L2 switch function to 8 ports of them with 1000BASE speed. The other SFP+ ports are unused to save hardware resources. The board also has TCAM/DRAM daughter board. The TCAM daughter board has 8 TCAM (model: IDT75K72100) chips and is configured as 144bit x 1,024K entries. The DRAM daughter board has 2 channeled DDR3-1600 SDRAM whose capacity is 16GB.

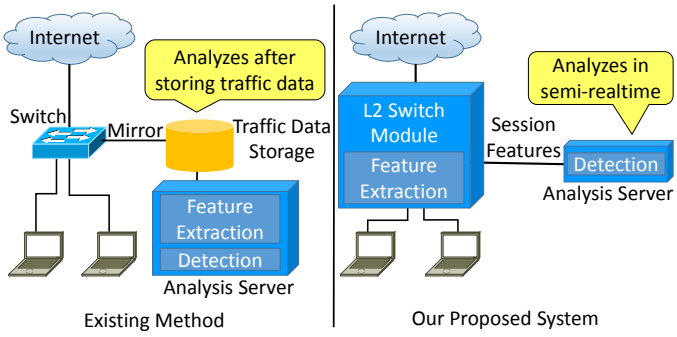

Figure 1. The difference between existing method and our method

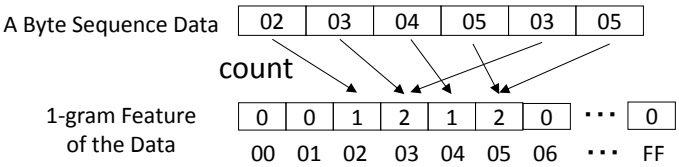

Figure 2. 1-gram Feature

We implemented feature extraction functionality [1]. When it receives a TCP packet, the Packet Feature Extraction Module extracts the header information and the 1-gram feature of the payload from the packet as shown in Figure 3. The header information contains the payload size and the src Internet Protocol(IP) address / src port number / dst IP address / dst port number, respectively. The 1-gram feature is the byte based value frequency of the payload as shown in Figure 2. Firstly, the payload is divided into 1-byte length and the counter for 1-gram feature counts appearance of 1 to 255 value in the playload. This feature used for PAYL detection algorithm, but it requires too many arithmetic resources.

Based on above packet based information extraction, our system cumulates them to create session features as the session continues. The features required to identify the session are shown as below.

- IP Address : Client / Server
- Port Number : Client / Server
- Total Packet Count : Each Communication Direction
- Total Payload Size : Each Communication Direction
- 1-gram Feature : Each Communication Direction
- Finish State : The State of Cumulation Termination

When cumulation of session features is finished, our system outputs them. There are several patterns to terminate the cumulation, so that we prepare the Finish State. It indicates three patterns of termination that are termination by FIN Flags, termination by RST Flags, and termination by lack of buffer capacity.

C. Data Structure for Session Feature Extraction

We utilize off-chip memory to record session features. One TCAM and DRAM entry of fixed size storage area are assigned for each session. When the session starts, these entries are assigned for cumulation. Until the session finishes, features are cumulated using the entries as a buffer of partial cumulation. After the session finishes, assigned entries are deleted by sending their content to the analysis server. The contents of TCAM and DRAM entries are as follows.

TCAM Entry
IPaddrLow(4bytes) , IPaddrLowPort(2bytes) ,
IPaddrHigh(4bytes) , IPaddrHighPort(2bytes) ,
IsIPaddrHighServer(1byte) ,
DRAMaddr(4bytes)

DRAM Entry
1-gramPayloadFeature (1,024bytes × 2)
PacketCnt (4bytes × 2)
PayloadSize (4bytes × 2)
SessionState (1byte)

TCAM entry works as an index of DRAM entry. A search key of TCAM entry consists of IPaddrLow, IPaddrLowPort,

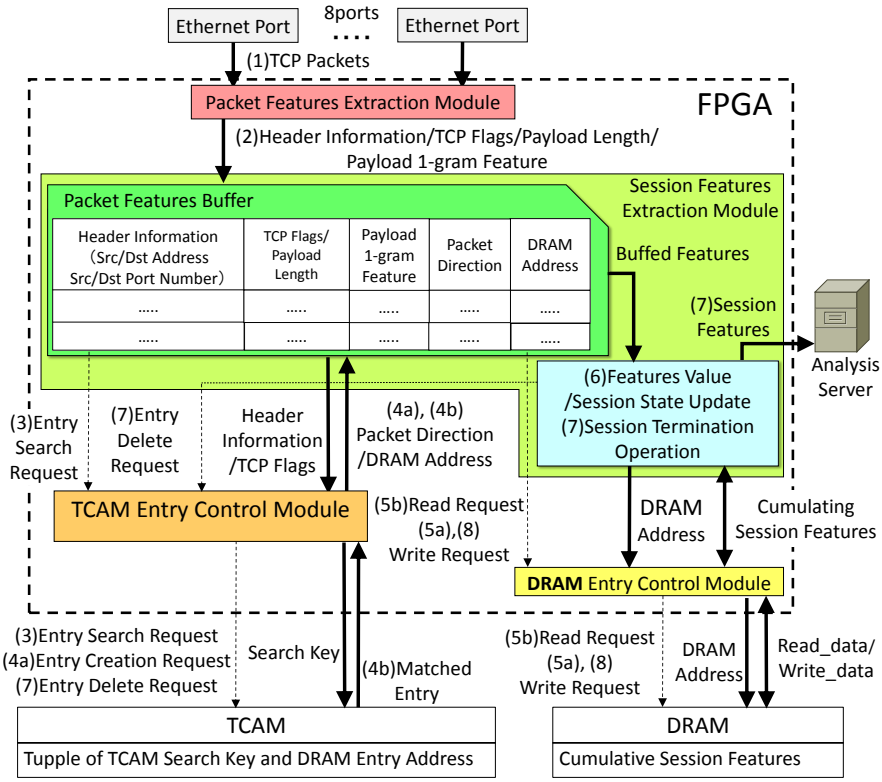

Figure 3. The behavior of Session Features Cumulation

IPaddrHigh, and IPaddrHighPort. IPaddrLow is the smaller one of either client IP address or server IP address in 32-bit numeric order. To save number of entries between bidirectional communication, we sorted IP addresses with 32-bit numeric order for index. By sorting this way, packets of both directions of a same session are assigned to the same entries. IPaddrLowPort is the port number of the IPaddrLow host and IPaddrHighPort is the port number of IPaddrHigh host. IsIPaddrHighServer is the identifier of the server which is required to identify the server after IP address sorting. With these 5 fields, we can identify the session. The DRAMaddr is the address of the DRAM entry which keeps detailed cumulating session features like 1-gram features of the session. In this way, we combine TCAM and DRAM for the session identification and the session features cumulation buffer. This organization can reduce consumption of TCAM and DRAM entries.

DRAM entry stores cumulating features while the session is in progress. The PacketCnt is the cumulation of packet count and the PayloadSize is the cumulation of payload size. The 1-gramPayloadFeature is the cumulation of 1-gram feature of all packets. These three fields are separated by communication direction.

D. The Operation of the System

Figure 4 shows a flowchart of the session feature cumulation process when a TCP packet arrives. Figure 3 shows a block diagram of the implementation which executes the session feature cumulation process. In our previous study, we implemented Packet Features Extraction Module as shown in Figure 3. In this study, we modified Session Feature Extraction Module and developed TCAM Entry Control Module, and DRAM Entry Control Module.

The operation of the system is as follows.

(1) When an Ethernet Port receives a TCP packet, the packet data comes into the system. Then, the Packet Features Extraction Module extracts header information, TCP flags, payload length, and 1-gram feature of the payload from the packet.

(2) It requires some latency to access the TCAM and the DRAM because of their access latency. Therefore, the packet features are buffered in Session Feature Extraction Module.

(3) The header information and TCP flags are sent to the TCAM Entry Control Module. Then, the module calculates the search key by comparing two IP addresses of the header information as unsigned 32-bit integer order and defines IsIPaddrHighServer by recording the information whether IP addresses are sorted or not. If the TCAM port is available, the module sends the search request to the TCAM.

(4) (4a) If the matched entry does not exist in the TCAM and the packet is a SYN packet, we treat it as a start of the session and the module generates new entry for the new session. The module calculates the address of the new DRAM entry and initializes the contents of the TCAM entry. Also, it generates the packet direction information. Both the DRAM address and the packet direction information are sent

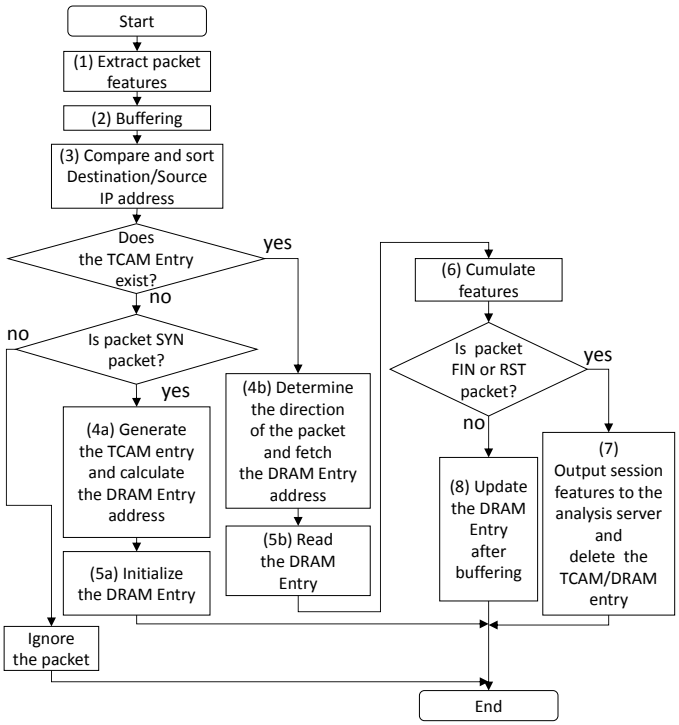

Figure 4. Operation Flowchart

to Session Features Extraction Module and recorded in the Packet Features Buffer.

(4b) If the matched entry exists, there is a session which is already started. The packet direction is determined by sorting and the IsIPaddrHighServer of the matched entry. Both the DRAM address and the packet direction are recorded in Packet Features Buffer.

(5) (5a) (comes from (4a)) If the session is the new session, the new session feature is written into the DRAM as a new entry through the DRAM Entry Control Module.

(5b) (comes from (4b)) Session Features Extraction Module sends a read request to the DRAM through the DRAM Entry Control Module, to read partially cumulated session features.

(6) After reading out the DRAM entry, the module cumulates the 1-gram payload feature to the 1-gramPayloadFeature, cumulates the payload length to the PayloadSize, and increments the PacketCnt. If the packet is a FIN or a RST packet, the following operation becomes (7). Otherwise, the following operation becomes (8).

(7) If the packet is a FIN or a RST packet, the session is finished with this packet. The cumulated session features are sent to the analysis server. Also the TCAM entry delete request is sent to the TCAM through TCAM Entry Control Module. This operation is also activated when the system consumes the entire TCAM and the DRAM entries, and it has to terminate the oldest cumulating session.

(8) The module updates the fields of the DRAM entry with newly cumulated values.

IV. Implementation Results

In this study, we implemented the following three modules, Session Features Extraction Module, TCAM Entry Control Module, and DRAM Entry Control Module by the Verilog Hardware description language(HDL). However, we have not finished the implementation of the function of entry deletion nor termination of oldest cumulating session when the system consumes all TCAM and DRAM entries.

We synthesized those modules with Altera Quartus II 13.1. Table I and II show the result of the current FPGA resource utilization. According to Table 1, the modules in this implementation occupy 42% of whole logic elements because we implemented 256 adders for cumulating 256 1-gram payload feature simultaneously. Therefore, logic elements of whole system are 96% of all logic elements, so that there is no space to optimize for operating clock frequency and no room for additional functions. Hence, we do not include operating clock

TABLE I. LOGIC ELEMENTS UTILIZATION OF COMPONENTS

Components	used	total	usage
Whole System	225,474	234,720	96%
The Three Modules	97,902	234,720	42%

TABLE II. REGISTERS UTILIZATION OF COMPONENTS

Components	used	total	usage
Whole System	438,482	938,880	47%
The Three Modules	204,909	938,880	22%

frequency results. Also, there is still unimplemented functionality, so that we have to improve our current implementation to reduce the usage of logic elements. According to Table II, the three modules occupy 22% of all registers because we implemented Packet Feature Buffer as a register array. If we increase the number of buffers to catch up with higher throughput (e.g., 10GBASE $\times$ 8), we have to re-implement it with block RAM. Furthermore, the current implementation accesses the DRAM when it receives a TCP packet. This becomes a possible bottleneck of the system, so that we are just considering some type of cache.

There are still many difficulties, but our system can currently handle a total of 1,024K entries of buffers. Therefore, it can handle 1,024K sessions simultaneously.

V. Throughput Estimation

We estimated the throughput of current implementation. In the worst case, the proposed system accesses the TCAM twice for searching and making a new entry. After that, it accesses the DRAM twice for reading and writing entries. In the proposed system, we made pipeline stages to enable accessing the TCAM and the DRAM in parallel, so that we only have to consider whether either of them is a bottleneck or not. Firstly, we estimate throughput from DRAM side because DRAM becomes bottleneck in many systems. DRAM access time for transmitting given data size (byte) is shown as

$$tRAS + tRCD + tCAS + tCLK \times \frac{data\_size}{8} \quad (1)$$

 ISBN: 978-1-61208-435-0

Where tRAS is row access strobe time, tRCD is row to column delay time, tCAS is column access strobe time, and tCLK is clock cycle time of data transfer. By substituting typical values of DDR3-1600 SDRAM and the data size for one session, the above formula is translated as follows.

$$45ns + 12.5ns + 12.5ns + 1.25ns \times \frac{2065}{8} = 393ns \quad (2)$$

Note that the above value is DRAM access time for one access. The proposed system requires two DRAM accesses in one packet processing, so that the substantial DRAM access time becomes twice that value. But our system has two DRAM channels, so that if we adequately interleave DRAM access, the DRAM throughput becomes twice that much value. Thus, DRAM access time per one packet processing becomes 393ns in our system. The DRAM access time is 393ns and packet throughput is 2.54Mpps (packet per second). The data throughput is related to the size of a packet and number of DRAM accesses per packet as follows.

$$\frac{packet\_throughput \times average\_packet\_size}{access\_count\_for\_one\_procedure} \quad (3)$$

So, if we assume 1500 bytes packets, the throughput becomes 38.1Gbps. If we assume 64 bytes short packets, the throughput becomes 1.62Gbps. Thus, we have to consider some filtering scheme for huge amount of short packets. On the other hand, the TCAM which we utilized can treat 250M search per second and it is much larger than that of the DRAM. So, the TCAM does not affect throughput in the proposed system.

VI. Conclusions and Future Work

We proposed the method of TCP session features extraction for anomaly-based Network IDS by FPGA using off-chip memories. We implemented the proposal to FPGA and confirmed that we can implement 1,024K session treatable system. But we also confirmed that the current implementation consumes almost all FPGA resources, so it requires further updating to implement additional functions and raise throughput.

In the future, we will reduce hardware resource consumption of implementation by modifying feature cumulation circuit to calculate in multi cycles. This alteration will enable us to implement additional functionality. Moreover, we will also improve the algorithm of the feature extraction processes to raise throughput of the system. Finally, we will evaluate the real throughput of the system by operating it in real traffic environment.

Acknowledgment

This research is aided by R&D of detective and analytical technology against advanced cyberattack, administered by the Ministry of Internal Affairs and Communications.

References

[1] S. Yanase, H. Shimada, Y. Yamaguchi, and H. Takakura, "Network access control by FPGA-based network switch using HW/SW cooperated IDS," TECHNICAL REPORT OF IEICE, vol. 114, no. 286, 2014, pp. 91–96.

[2] S. Yanase, H. Shimada, Y. Yamaguchi, and H. Takakura, "Implementation of FPGA section for anomaly detection acceleration by HW/SW cooperation (in Japanese)," TECHNICAL REPORT OF IEICE, vol. 114, no. 116, 2014, pp. 75–80.

[3] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in Recent Advances in Intrusion Detection. Springer, 2004, pp. 203–222.

[4] Snort, "Snort.Org," http://www.snort.org, Accessed: 2015-8.

[5] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in GI/ITG Workshop MMBnet, 2007.

[6] R. Perdisci, G. Gu, and W. Lee, "Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems," in Data Mining, 2006. ICDM'06. Sixth International Conference on. IEEE, 2006, pp. 488–498.

[7] T. Katashita, Y. Yamaguchi, A. Maeda, and T. Kenji, "FPGA-based intrusion detection system for 10 gigabit ethernet," IEICE transactions on information and systems, vol. 90, no. 12, 2007, pp. 1923–1931.

 ISBN: 978-1-61208-435-0