Behavior Signature for Big Data Traffic Identification

Sung-Ho Yoon, Jun-Sang Park and Myung-Sup Kim
Dept. of Computer and Information Science
Korea University
Korea
{sungho_yoon, junsang_park, tmskim}@korea.ac.kr

ChaeTae Lim and JunHyung Cho
Korea Internet Security Center
Korea Internet & Security Agency
Korea
{scorch, chtim}@ kisa.or.kr

Abstract—**With the rapid development of the Internet and the popularization of multimedia services, Internet traffic has become a big data traffic that its volume, variety and velocity has dramatically increased. This phenomenon causes several limitations in traffic classification such as increased computational complexity and difficult real-time control. In this paper, we propose a behavior signature for application-level traffic identification to overcome these limitations. The proposed behavior signature is the identity pattern of traffic behavior appearing in the first few request packets of plural traffic flows when a specific function is conducted by an application. This is in contrast to the previous signature techniques that usually use a singular packet or flow for feature extraction and traffic identification. In order to prove the feasibility of the proposed behavior signature, we present the experimental results based on five popular applications.**

Keywords—behavior signature; big data analysis, traffic classification; traffic identification; network management

I. Introduction

The efficient network management is emphasized with the rapid growth in Internet penetration and greater diversification of Internet applications [1][2]. An application-level traffic identification that ascertains which application is contributing to the network traffic should be preemptively accomplished before applying the various network management policies. The final goal of application-level traffic identification is to accurately name all of the Internet network traffic according to the corresponding applications. To utilize the identification results as traffic control policies effectively, the identification process should be completed within a given time. Therefore, the Internet traffic identification is a time sensitive process.

However, the Internet traffic has become big data having increasing volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources) due to the rapid development of Information & Communication Technology (ICT), which includes Internet, mobile, and mass media [3]. The many previous identification methods that are mostly focusing on the completeness and accuracy have limitations when applied to real operational network due to the explosion of traffic. Typical limitations include high computational complexity and difficult real-time control [4]. Although there are several big data analysis solutions such as Hadoop Distributed File Systems (HDFS), cloud technology, and hive database, a fundamental approach that is identification method using various traffic features is needed to solve these problems for real-time identification of big traffic data.

In this paper, we propose a behavior signature and an extraction algorithm to overcome the limitations of earlier methods. Most Internet applications generate multiple traffic flows when a user performs a specific function such as login, chat, file transfer, etc. For example, several flows that are involved in authorization, application update, and encryption negotiation are generated in the log-in phase of most applications. Moreover, there is a unique pattern in the sequence and interval of these flows. Thus, we devise the behavior signature centered on the idea that the unique patterns of several flows can represent a specific function.

The behavior signature uses the inter-flow unit to extract the signature and identify traffic which is contradictory to packet or flow units used in the previous signature technique. The inter-flow unit is a set of the first request packets of more than one flow. This new traffic unit has some advantages. It is possible to identify several flows at once (more than one flow) and it can be applied immediately after a specific function has occurred (first request packets). Another distinguishing characteristic of the signature is that it uses combinations of various traffic features such as IP address, port number, L4 protocol, and payload data. It allows that we can extract signatures easily because using multiple features can expand extraction range compared to using single feature only. In addition, the behavior signature can reduce the identification time because all features, even payload, are located in fixed offset. We use the first *N* bytes payload in the request packet.

The remainder of this paper is organized as follows. We survey several existing behavior based on traffic identification methods in Section 2. In Section 3, we describe the concept of the proposed behavior signature in detail. In Section 4, we propose the automatic extraction algorithm of the behavior signature. In Section 5, we discuss the experimental results to prove the feasibility of behavior signature. Lastly, we present our conclusion and future work in Section 6.

II. Related Work

Because of Internet traffic growth and development of various Internet applications, several studies on traffic identification have been proposed already. The earlier studies have suggested that a port number was sufficient to identify traffic because most applications follow the Internet Assigned Number Authority (IANA) assertion [5][6]. However, recently it is becoming more difficult to identify traffic despite using

This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

978-1-4799-3919-0/14/$31.00 ©2014 IEEE

various traffic features such as payload [7][8], statistical information [9][10] because of the growth of traffic complexity. This situation is the result of network managers blocking traffic of unwanted applications while the application developers try to bypass the manager's network policy. For this reason, the behavior based on traffic identification methods [11] that analyze the unique behavior pattern of applications have been proposed more often than methods that use only one particular feature such as port or payload. This section examines the existing behavior based on traffic identification methods and their limitations.

T. Karagiannis et al.[11] proposed the method that classifies the traffic generated by a host according to three levels (social, functional, and application). This method was very simple and easy to use in diverse networks and could identify traffic without using port numbers and payload data. One limitation, however, was that this method could not distinguish applications generated in a single host. This was due to the assumption that the traffic on a single host was generated by a particular application.

L.Bernaile et al.[12] analyzed the traffic using first K packet size distribution (K-data-packet-size) for application-level traffic identification in real time. Although this method solves the problem about the invasion of privacy by using packet size and direction without inspecting payload, it was limited for it was difficult to identify traffic between applications that implement that the same protocol because they have similar statistical characteristics.

In this paper, we propose the behavior signature using the first request packets of several flows generated when an application executes a particular function. Because the used features are extracted from several flows rather than a single flow, it is easy to find unique pattern. In addition, by using header information and the N bytes string located in first part of payload, we solve the problems with computational complexity and invasion of privacy.

III. Behavior Signature

In this section, we discuss various traffic identification units and traffic features that are used as signature attributes for defining the proposed behavior signature.

The traffic features that are used in the behavior signature are the destination IP address, destination port, L4 protocol, and the first N bytes in the first payload packet of a flow. The signature consists of a combination (we call this as entry) of these features. Because of this characteristic, it is convenient to extract the signature compared to using a single feature. Header information such as IP address, port, and protocol has significant meaning in the server-client model and using fixed port traffic. Payload information has been used as a salient key for identifying traffic; however, because of the invasion of privacy, its usage has declined. To solve these privacy issues, we use the first N bytes payload only rather than full payload. It is easy to resolve the privacy issues even with low computational complexity because the signature uses fixed offset and length bit string in payload.

Figure 1 shows the various traffic units for traffic identification such as packet, flow, and inter-flow. The packet unit uses the header information and payload in a single packet. It is good at real-time control because of its ability to control unwanted traffic after inspecting packet immediately. However, it is hard to extract signature because the range of extraction is relatively small. In addition, the significant overhead is caused by inspecting all packets to identify traffic. The flow unit, on the other hand, uses not only packet attributes but also additional information such as inter-arrival time, packet size distribution, total size of bytes, etc. The flow unit is good for extracting signatures because of a relatively large range of extraction. In other words, more attributes can be utilized for signature extraction in the flow unit compared to the packet unit. However, it has limitations such as low accuracy and real-time control. The flow unit can be used to identify traffic after the flow is over.

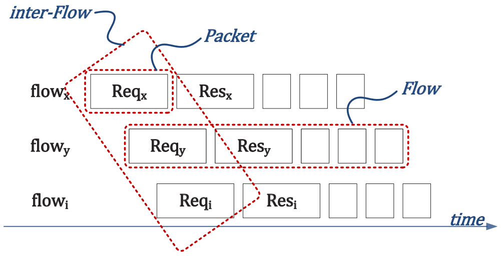


Fig. 1. Various traffic units for traffic identification

The behavior signature is applied in the inter-flow unit, and it seeks to minimize the limitations of the packet and flow units and maximize their advantages. The inter-flow is a set of first request packets of several flows. Therefore, it uses not only packet unit attributes but also sequence and interval information. It is easy to create the signature because the range of extraction is large by the use of plural flows. In addition, inter-flow has the ability to control traffic in real-time because traffic is identified by the signature in the first request packet located at the beginning of flow. Although the inter-flow unit has many advantages, it also has disadvantages. The inter-flow unit operates under the assumption that plural flows occur when a single function is performed. If a single function makes a single flow, the behavior signature does not apply. We leave this limitation to be addressed in our future work.

The behavior signature consists of several entries having aforementioned attributes of traffic. The following equations define the behavior signature respectively.

$$BS = \left\{ \begin{matrix} A, T, I, E_1, E_2, E_3, \dots, E_n | \\ n \geq 2, \\ Src(E_1) = Src(E_2) = \cdots = Src(E_n) \end{matrix} \right\} \quad (1)$$

$$E = \{2^F | F = \{ip, port, prot, payload\}, E \neq \emptyset\} \quad (2)$$

Behavior signature (BS) consists of application name (A), type (T), interval (I), entries (E_n) where $n \geq 2$. Entry (E) is a power set of destination IP address (ip), destination port ($port$), L4 layer protocol ($prot$), and first N bytes payload ($payload$), where null set is excluded. i.e., we selectively use the features of entry in traffic identification if the selected feature has the meaning. For example, we exclude destination IP address and destination port from entry when the application uses random ports under P2P connection. In case, we write "any" in ip

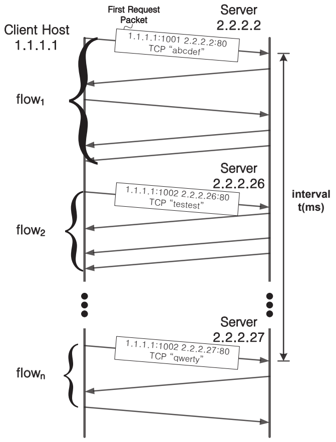
attribute. The source host ($Src(E_x)$) of all entries is the same because the behavior signature represents the behavior of single function traffic from a single host.

Table I explains the attributes of the behavior signature. Application name (*A*) is used for naming identified traffic. The entry applying method Type (*T*) is either Sequence (*Seq*) or Set (*Set*). *Seq* means that identification is conducted in serial order whereas *Set* means that identification is done randomly in a particular time interval. Interval (*I*) is the period of time in milliseconds (ms) from the first entry to the last entry, i.e., the time of applying all entries of the behavior signature.

TABLE I. BEHAVIOR BASED SIGNATURE ATTRIBUTES AND EXPLANATION

Attribute		Explanation
A		Application name
T		Entry applying method Sequence (*Seq*), Set (*Set*)
I		Interval applying all entries (ms)
E	*ip*	Destination IP address in CIDR notation
	port	Destination port number
	prot	L4 protocol (TCP, UDP)
	payload	First *N* bytes payload HTTP : first 10 bytes Non-HTTP : first 2 Bytes
Src(Ex)		Source IP address of Entry x

The Entry (*E*) consists of destination IP address (*ip*), destination port number (*port*), L4 layer protocol (*prot*), and the first *N* byte payload (*payload*). Destination IP and port are the destination of traffic that will be identified by the entry. The IP address is represented in Classless Inter-Domain Routing (CIDR) notation. The L4 layer protocol is either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) when transmitting traffic on the Internet. We use the first *N* bytes payload rather than full payload. Depending upon whether there is HTTP or non-Http traffic, we use the first ten bytes payload to distinguish method of HTTP such as GET, POST, and PUT, etc. or its first two bytes respectively.



A		X
T		Seq
I		t
E1	ip	2.2.2.2
	port	80
	prot	TCP
	payload	"ab"
E_2	ip	2.2.2.26
	port	80
	prot	TCP
	payload	"te"
...		
E_N	ip	2.2.2.27
	port	80
	prot	TCP
	payload	"qw"

Fig. 2. Example of behvior signature

Figure 2 is an example of the behavior signature. When application *X* generates *N* flows within time interval *t*, we can extract the signature with *N* entries. Each entry has features of the first request packets of each flow.

IV. EXTRACTION ALGORITHM

This section describes the extraction algorithm for the behavior signature. It consists of the first request packet extraction module, the candidate signature extraction module, and the signature selection module.

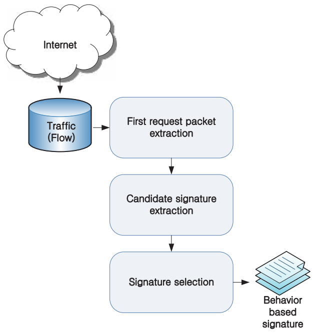


Fig. 3. Extraction algorithm for behavior based signature

Figure 3 is a detailed flow diagram of the extraction algorithm. From the input traffic, we extract the first request packet and then extract all candidate signatures from every conceivable combination of the entries. Finally, we select the behavior signature from the candidate signatures.

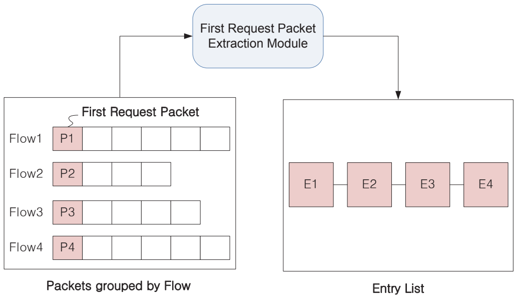


Fig. 4. Input-output data of the first request packet extraction module

The first request packet extraction module composes an entry list from the first request packet of input traffic. This entry list is sorted by occurrence time of the packet.

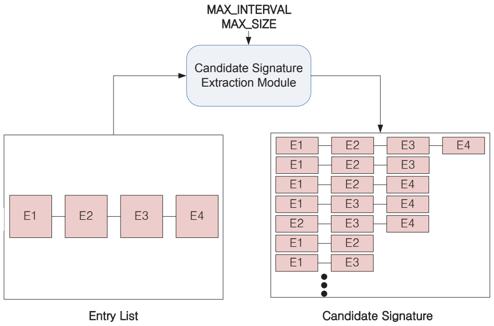


Fig. 5. Input-output data of the candidate signature extraction module

TABLE II. RESULT OF BEHAVIOR BASED SIGNATURE EXTRACTION AND EXAMPLE

Application	Num. of signature	Example
Nateon	48	{Nateon, Seq, 4324, (203.xxx.xxx.91/32, 5004, 6, "PVER 1 4.1.2485 5.0"), (120.xxx.xxx.0/24, 5004, 6, "NCPT 1"), (117.xxx.xxx.17/32, 80, 6, "GET /keyword37_u2.op"), (203.xxx.xxx.117/32, 80, 6, "POST /client/club/Ge"), (211.xxx.xxx.0/24, 80, 6, "GET /upload/notice/"), (211.xxx.xxx.0/24, 80, 6, "GET /upload/"), (211.xxx.xxx.0/24, 80, 6, "GET /upload/"), (211.xxx.xxx.0/24, 80, 6, "GET /upload/"), (117.xxx.xxx.12/32", 80, 6, "GET /nateon/ticker H"), (120.xxx.xxx.20/32, 80, 6, "POST /client/CountMe")}
DropBox	1	{DropBox, Seq, 3258, (any, 443, 6, "0x16 0x03 0x01 0x00 0x5B 0x01 0x00 0x00 0x57 0x03 0x01 0x50"), (any, 80, 6, "GET /subcribe?host_")}
UTorrent	7	{UTorrent, Set, 5000, (any, any, 17, "d1:ad2:id20:"), (any, any, 17, "A."), (any, any, 17, "d1:ad2:id20:")}
Skype	3	{Skype, Seq, 5000, (any, any, 6, "GET /ui/0/5.10."), (any, any, 6, "0x16 0x03 0x01 0x00")}
Teamviewer	1	{Teamviewer, Seq, 4991, (any, 5938, 6, ".$"), (any, 5938, 17, "0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00")}

The candidate signature extraction module creates the candidate patterns from every conceivable combination after entering the entry list. Because of high computational complexity, we set the thresholds such as the maximum interval from the first entry to last entry (MAX_INTERVAL) and the maximum entry size (MAX_SIZE); namely candidate signatures are extracted by limiting MAX_INTERVAL and MAX_SIZE.

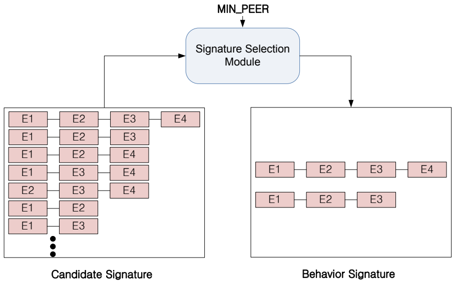


Fig. 6. Input-output data of the signature selection module

The signature selection module chooses the behavior signature to be the one with the host counts exceeding the minimum peer count using the signature (MIN_PEER) from every possible candidate signature. The behavior signature is the most commonly used pattern when all hosts use a particular application.

V. EXPERIMENT AND RESULT

This section details the experimental results and feasibility of the behavior signature using five popular applications. We select the five popular applications - Nateon: messenger; DropBox: file hosting; UTorrent: P2P file transfer; Skype: messenger; Teamviewer: remote desktop - as the target applications. To test performance accurately, we collected the traffic data by conducting various functions on four different hosts at two particular times.

A. Signature Extraction

Table II gives the results of signatures selected by using the proposed algorithm. This test set MAX_INTERVAL to 5,000ms, MAX_SIZE to 10, and MIN_PEER to 4.

In the case of Nateon, many signatures, 48, were extracted because it has complicated traffic. In particular, Nateon communicates with authorization, update, pop-up, and main server at the log in phase. Because Nateon is operated under the server-client model and uses fixed port, the signatures that were extracted from this application have all the attributes as shown in Table I. The example, located in right column of Table II, means that if the ten entries are matched to ten first request packets of flows in serial order during the interval of 4,324ms, the flows are identified as Nateon.

In the case of UTorrent, seven signatures were extracted. Thus, we mark destination IP address and port number as “any” because this application operates under P2P and uses random port. As shown in Table II, if the two entries are matched to two first request packets of flows in the given interval in random order, the flows are identified as UTorrent.

B. Performance Evaluation

We measure the accuracy (precision, recall) of the proposed signature method by using the mixture traffic of the five applications being considered. The following equations measure precision and recall respectively.

$$Precision = \frac{\text{TP}}{(\text{TP+FP})} \quad (3)$$

$$Recall = \frac{\text{TP}}{(\text{TP+FN})} \quad (4)$$

TABLE III. ACCURACY OF BEHAVIOR BASED SIGNATURE

Application	Unit	Precision	Recall
Nateon	flow	1.00 (447/447)	0.60 (447/741)
	byte(K)	1.00 (5064/5064)	0.02 (5064/254110)
DropBox	flow	1.00 (193/193)	0.78 (193/247)
	byte(K)	1.00 (5303/5303)	0.15 (5303/35708)
UTorrent	flow	1.00 (2999/2999)	0.17 (2999/18106)
	byte(K)	1.00 (2741745/2741745)	0.66 (2741745/4182441)
Skype	flow	1.00 (127/127)	0.06 (127/2088)
	byte(K)	1.00 (1589/1589)	0.02 (1589/103342)
Teamviewer	flow	1.00 (239/239)	0.63 (239/385)
	byte(K)	1.00 (8237/8237)	0.04 (8237/215845)
Total	flow	1.00 (4005/4005)	0.18 (4005/21487)
	byte(K)	1.00 (2761938/2761938)	0.57 (2761938/4791446)

TABLE IV. COMPARISON COMPLETNESS BETWEEN PAYLOAD AND BEHAVIOR BASED SIGNATURE

Application	Unit	Completeness					
		PS	BS	$\mathrm{PS}\cup\mathrm{BS}$	$\mathrm{PS}\cap\mathrm{BS}$	$\mathrm{PS}^c\cap\mathrm{BS}$	$\mathrm{PS}\cap\mathrm{BS}^c$
Nateon	flow	0.73 (543/741)	0.60 (447/741)	0.73 (543/741)	0.60 (447/741)	0.00 (0/741)	0.13 (96/741)
	byte(K)	0.93 (235,143/254,110)	0.02 (5,064/254,110)	0.93 (235,143/254,110)	0.02 (5,064/254,110)	0.00 (0/254,110)	0.91 (230,079/254,110)
DropBox	flow	0.26 (64/247)	0.78 (193/247)	0.78 (193/247)	0.26 (64/247)	0.52 (129/247)	0.00 (0/247)
	byte(K)	0.01 (68/35,708)	0.15 (5,303/35,708)	0.15 (5,303/35,708)	0.01 (68/35,708)	0.15 (5,234/35,708)	0.00 (0/35,708)
UTorrent	flow	0.79 (14,358/18,106)	0.17 (2,999/18,106)	0.80 (14,488/18,106)	0.15 (2,869/18,106)	0.01 (140/18,106)	0.63 (11,489/18,106)
	byte(K)	0.96 (4,020,339/4,182,441)	0.66 (2,741,745/4,182,441)	0.99 (4,171,534/4,182,441)	0.62 (2,578,702/4,182,441)	0.04 (163,043/4,182,441)	0.34 (1,429,789/4,182,441)
Skype	flow	0.02 (44/2,088)	0.06 (127/2,088)	0.06 (127/2,088)	0.02 (44/2,088)	0.04 (83/2,088)	0.00 (0/2,088)
	byte(K)	0.01 (51/103,342)	0.02 (1,589/103,342)	0.02 (1,589/103,342)	0.01 (51/103,342)	0.01 (1,538/103,342)	0.00 (0/103,342)
Teamviewer	flow	0.01 (1/385)	0.62 (239/385)	0.62 (240/385)	0.00 (0/385)	0.62 (239/385)	0.01 (1/385)
	byte(K)	0.01 (1/215,845)	0.04 (8,237/215,845)	0.04 (8,239/215,845)	0.00 (0/215,845)	0.04 (8,237/215,845)	0.01 (1/215,845)
Payload Signature (PS), Behavior Signature (BS)							

A True Positive (*TP*) of application X means the proportion of X traffic identified as X correctly. Otherwise, a False Positive (*FP*) of application X means the proportion non-X traffic identified as X incorrectly. False Negative (*FN*) of application X means the proportion of X traffic identified as non-X incorrectly. Thus, precision is the ratio of clearly identified traffic to the total identified traffic, and recall is the ratio of clearly identified traffic to the application traffic.

Table III shows the accuracy (precision, recall) of the behavior signature about of the five applications. All signatures identify the traffic precisely, i.e., precision is 1.00 in all applications, and this is because the signatures were extracted from several hosts. In case of recall, it depends on the applications. The average is 0.18 in flow units and 0.57 in byte units. This is caused by the statistical characteristics of each application as having heavy or light flow. Thus, the behavior signature is more useful in detection and control of applications than traffic monitoring.

C. Comparison with payload Signature

We conduct a comparison test on between the payload signature method and the behavior signature method. We use the payload signature based on the Longest Common Subsequence (LCS) algorithm [13]. The payload signatures used in this test are shown in Table V.

TABLE V. PAYLOAD SIGNATURE FOR COMPARISON TEST

Application	No. Signature	Example
Nateon	42	.*naeon\.nate\.nate\.com.* ^PVER.*
DropBox	3	^GET /subscribe.host_int=.* .*Dropbox,Inc.*dropbox\.com.*
UTorrent	13	/*BitTorrent protocol.* .*d1:ad2:id20.*
Skype	1	.*User-Agent:.*Skype.*
Teamviewer	1	^..\x00\x17\x24\x6A.\x00.*

Table IV shows the results of behavior and payload signatures. The following equation is the metric for measuring performance.

$$Completeness = \frac{\text{Identified Traffic}}{\text{Total Traffic}} \quad (5)$$

PS is the ratio of identified traffic using the payload signature. BS is the ratio of identified traffic using behavior signature. $\mathrm{PS}\cup\mathrm{BS}$ is the ratio of total identified traffic using either payload or behavior signatures. $\mathrm{PS}\cap\mathrm{BS}$ is the ratio of overlap of traffic identified by the payload and behavior signature. $\mathrm{PS}^c\cap\mathrm{BS}$ is the ratio of traffic identified using the behavior signature, but not identified when using the payload signature. $\mathrm{PS}\cap\mathrm{BS}^c$ is inverse case.

The value of $\mathrm{PS}^c\cap\mathrm{BS}$ for Nateon is zero because the traffic identified by using the payload method includes all traffic by behavior method. This is due to the Nateon application characteristic using the open protocol instead of the encryption of traffic. On the other hand, the behavior signatures of Dropbox, Teamviewer, and Skype include the payload signature. In the case of Dropbox, HTTPS traffic is used for data encryption. Therefore, it is difficult to extract the payload signatures using the LCS algorithm. The behavior signature method, however, can extract the signature using a combination of several entries to identify traffic precisely.

According to this comparison test, we can find that payload and behavior signatures are complementary relation in traffic identification. In case of using the open protocol such as Nateon and UTtorrent, the payload signature has good performance. In case of using the encryption and proprietary protocol such as DropBox, Skype, and Teamviewer, behavior signature is good for it.

Figure 7 shows a comparison test of execution time. *T(PS)* is the execution time when it is applied to given test traffic by payload signature. *T(BS)* is the execution time when it is applied by behavior signature. *T(BS+PS)* means the execution time when the behavior signature is applied at first, and the payload signature is applied just in case of unidentified traffic. Although there is distinct differences about the execution time in each application caused by difference of the amount of traffic and the number of signature, *T(PS)* is longer than *T(BS)* commonly. *T(BS+PS)* is longer than *T(BS)* and shorter than *T(PS)* while it retains the completeness of PS∪BS. According to this test, we can find that the behavior signature is superior to the payload signature in execution time. In addition, when we use the behavior signature as a supplementary method for payload signature, it improves the execution time and completeness.

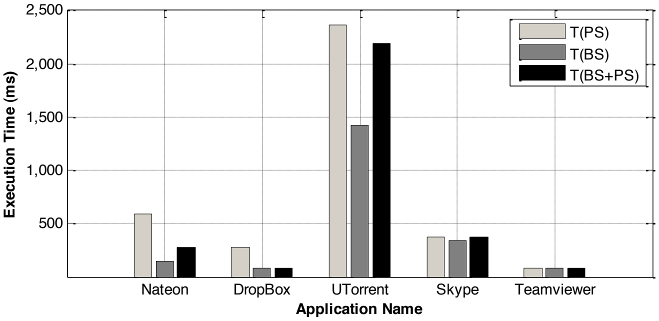


Fig. 7. Comparison execution time between payload and behavior signature

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a behavior signature and an automatic extraction method using the first request packets of multiple traffic flows when a single function executes to identify the big data traffic. This signature overcomes the limitation of previous methods using packet and flow units. We use the five popular applications to prove the feasibility of the proposed signature. Although our method shows low recall, the precision was 100% for all applications, and that means all extracted signature identified traffic correctly. A comparison test on the payload signature method proved that the behavior signature can be utilized as a supplementary method to identify the encrypted traffic flows. It improves the performance in execution time and completeness.

For future research, we plan to improve the extraction algorithm by applying various networks and applications. Moreover, we plan to develop the identification system based on the proposed signature to operate in real networks. And we are going to address the problem that a single function of application is conducted on a single flow by dividing the flow into multiple sup-flows according to the request-response of packets.

REFERENCES

[1] S.-H. Yoon and M.-S. Kim, “A study of performance improvement of internet application traffic identification using flow correlation,” *J. KICS*, vol. 36, no. 6, pp. 600-607, May 2011.

[2] S. Sen and J. Wang, “Analyzing peer-to-peer traffic across large networks,” in *Proc. Internet Measurement Conf. (IMC)*, pp. 137-150, Marseille, France, Nov. 2002.

[3] Douglas, Laney. "3D Data Management: Controlling Data Volume, Velocity and Variety," Gartner, Feb. 2001.

[4] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on internet traffic identification,” *IEEE Commun. Surveys Tutorials*, vol. 11, no. 3, pp. 37-52, July 2009.

[5] IANA, *IANA port number list*, Retrieved 5, 24, 2013, from http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml.

[6] J. Zhang and A. Moore, “Traffic trace artifacts due to monitoring via port mirroring,” in *Proc. End-to-End Monitoring Techniques and Services (E2EMON)*, pp. 1-8, Munich, Germany, May 2007.

[7] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, “Lightweight, payload-based traffic classification: an experimental evaluation,” in *Proc. IEEE Int. Conf. Commun (ICC) '08*, pp. 5869-5875, Beijing, China, May 2008.

[8] J.-S. Park, S.-H. Yoon, and M.-S. Kim, “Software architecture for a lightweight payload signature-based traffic classification system,” in *Proc. 3rd Int. Conf. Traffic Monitoring and Analysis (TMA) '11,* pp. 136-149, Vienna, Austria, Apr. 2011.

[9] K. Xu, Z.-L. Zhang, and S. Bhattacharya, “Profiling internet backbone traffic: behavior models and applications,” in *Proc. ACM SIGCOMM 2005*, pp. 169-180, Philadelphia, U.S.A., Aug. 2005.

[10] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” in *Proc. ACM SIGMETRICS,* pp. 50-60, Banff, Canada, June 2005.

[11] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, “BLINC: multilevel traffic classification in the dark,” in *Proc. ACM SIGCOMM 2005*, pp. 229-240, Philadelphia, U.S.A., Aug. 2005.

[12] L. Bernaille, R. Teixeira, and K. Salamatian, “Early Application Identification,” In The 2nd ADETTI/ISCTE CoNEXT Conference, Lisboa, Portugal, December 2006.

[13] Byung-Chul Park, Young J. Won, Myung-Sup Kim, James W. Hong, "Towards Automated Application Signature Generation for Traffic Identification," Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS) 2008, Salvador, Bahia, Brazil, Apri. 7-11, 2008, pp. 160-167.