This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

Network Application Identification Using Transition Pattern of Payload Length

Shinnosuke Yagi*, Yuji Waizumi*, Hiroshi Tsunoda* Abbas Jamalipour‡ Nei Kato† and Yoshiaki Nemoto*
* † Graduate School of Information Sciences
Tohoku University, Sendai, Japan
Email: * {yagi,wai,tsuno,nemoto}@nemoto.ecei.tohoku.ac.jp † kato@it.ecei.tohoku.ac.jp
‡ School of Electrical & Information Engineering,
University of Sydney, Australia
Email:a.jamalipour@ieee.org

***Abstract*—In recent years, information leakage through the Internet has become a new social problem. Many information leakage incidents are caused by illegal applications such as Peer-to-Peer (P2P) file sharing software. To prevent information leakage, early detection and blocking of the traffic exchanged by illegal applications is strongly required. In this paper, we propose a method for application discrimination of monitored traffic based on the transition pattern of payload length during start up phase of the communication. The proposed method does not need port numbers, which can be spoofed easily. Through experiments using real network traffic, we show that the proposed method can quickly and accurately discriminate applications.**

I. Introduction

In recent years, information leakage through the Internet has become a serious social problem. In 2005, more than 130 information leakage incidents have been reported in Japan [1]. Especially, information leakages due to setting errors of Peer-to-Peer (P2P) file sharing applications and illegal applications activated by Trojan horses are increased. To tackle this problem, it is important to identify applications and block flows transmitted from illegal applications.

The simplest application identification method utilizes port numbers. This method is based on an assumption that applications can be mapped to corresponding port numbers. However, this method is not effective because port number setting for applications can be change easily [7], [13]. To avoid this detection, applications select port number randomly or use port numbers for commonly used applications such as http, ftp and so on. For instance, if a P2P application uses port 80, the port number based method identify its flows as http flows.

To solve the port number interpolation problem, some studies to identify applications without port numbers have been conducted [6], [9], [14], [15], [21]. These techniques identify applications of observed flows based on the statistical information of these flows such as the number of packets, mean packet size and mean arrival time interval of packets for instance. However, these techniques need a lot of packets to obtain enough information to identify applications, and their identification accuracy is not enough.

In this paper, we point out that identification accuracy would decrease as the number of considered packets increase due to the large variance of the contents such as HTML files of HTTP. We therefore propose an identification technique using transition pattern of the inverse of payload length of packets. By using the inverse of the payload length, the proposed identification method can defuse the performance decrement of the identification caused by the large variance of the contents. Through experiments, we show that the proposed method can identify applications quickly and accurately.

II. Related work

As traffic classification technique not using port numbers, payload based approach has been proposed [5], [16], [18]. This approach examines payload of each packet to determine whether the payload contains specific characteristic signatures. This approach works well to identify network applications including P2P file sharing applications. However, this method hardly works when payload is encrypted. In addition, the payload examination can cause privacy problems.

Traffic identification techniques which are based on the behavior of hosts are proposed [4], [8], [19]. These techniques focus behavior of each host, such as how many hosts are connected and how many port numbers are used. Although they can detect specific running applications, they cannot identify flows which are transmitted by the applications. Consequently, these techniques cannot block flows transmitted from applications in violation of network management policies.

Other methods which use flow statistics are proposed. [6], [9], [14], [15], [21]. These methods are based on statistical information of the flow such as the number of packets, total data size transported in the flow, mean packet size and mean packet arrival time interval for instance. These method can identify each application flows not using port numbers or payloads. But they need a lot of packets or flows to obtain significant information for the identification. It is difficult for these methods to identify applications in the early stage of their flow and to block these flows before flows are finished.

To identify applications at the early stage of their connection, an approach using first few packets of each flow has been proposed [2], [3], [10], [20] These methods analyze first few packets of a flow. We have proposed a method using the character code histograms of the first few packets of a flow [20]. This method can identify applications before flows are

1525-3511/08/$25.00 ©2008 IEEE

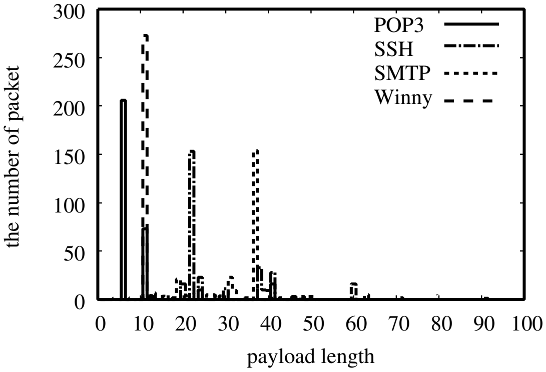

Fig. 1. Histogram of payload length

finished. However, since this method uses payload information of packets, it can hardly identify applications if payloads are encrypted. On the other hand, a method proposed in [3] is based on payload length and direction of the first five packets of a flow. This method does not use the payload information. However if there are two applications whose characteristics of the first five packets are similar, the methods can not identify the applications by the first five packets. Thus these methods need more packets to identify applications but the number of packets needed for accurate identification is not clear. This is a severe problem.

For practical use, identification methods should handle many kinds of applications. As the number of applications increases, the similarity among flows of different applications can be higher. To obtain enough information to identify, identification methods need to consider a lot of packets to find the differences between similar flows of different applications. But considering many packets causes a problem which can prevent accurate identification of other applications. In this paper, we show a problem which caused by considering more packets to identify applications, and propose an approach using inverse of payload length to solve this problem.

III. Transition pattern of payload length for application identification

Most applications have unique interactions of messages in the beginning of flows, such as user authentication, software version, available options and so on. These interactions are defined by their application protocols. This means that flows transmitted by a same kind of application have packets containing the same kind of information in the same order. For instance, we show distribution of payload length in Fig.1. We draw histograms by considering payload length of the first packet in each flows. We considered 300 flows of POP3, SSH, SMTP, Winny. Fig.1 shows that payload length of the first packet in flow has trend and the trend vary by applications.

Thus, we assume that same application flows have a similar transition pattern of the payload length of each packet in the first stage of the flows. Note that flow is the aggregated packets in a TCP connection.

A. Vector of transition pattern of payload length

To evaluate the transition pattern of payload length, we define vector $\vec{v}$ as

$$\vec{v} = [v_1, v_2, ...v_n]^T, \tag{1}$$

v_i is payload length of i th packet ($i = 1, 2, ..., n$). Dimension of $\vec{v}$ is the number of considered packets. To distinguish direction of packets, we define that the payload length of packets from host which sent SYN packet is positive value, and that of the other packets is negative value. To consider only application-layer information, we ignore packets with no payload (e.g. ACK packet which has no payload).

We use Euclidean distance between $\vec{v}$s to estimate similarity between flows.

B. Similarity of transition pattern

We assume that distance between $\vec{v}$s of a same application is small. To confirm whether this assumption is appropriate, we visualized distribution of $\vec{v}$s of each application by the Self Organizing Map (SOM) [11]. SOM is a subtype of artificial neural networks. It is trained using unsupervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space.

In this experiment, we capture flows of following eight kinds of applications: http, smtp, ssh, pop3, imap, pop3 over ssl (pops), http over ssl(https) and rtsp from a LAN, which is consisted of about 50 hosts. We also use flows of Winny and Share, which are P2P applications cause serious information leakage incidents in Japan. Flows of these two applications are captured from an experimental network which is consisted of 10 hosts.

Fig.2 shows the result when the dimension n is five. The color of the cell shows the distance between the adjoining cell, becomes darker when the distance is longer. A cell which has two labels means $\vec{v}$s of more than two kinds of applications placed to the cell. From this figure, we can confirm that there exist clusters of each application. The SOM model which is used to make this figure is 16x16 array of neurons.

Fig.3 depicts the result when the dimension n is ten. This figure shows that flows of some different applications belong to a cell and their distributions are overlapped each other. This overlapped distributions are caused by transmission of contents, such as HTML files. The packet payloads transmitting contents can vary enormously. Consequently, the size of the packets of contents also vary largely. This large variance of contents packets can be a notation of identification error.

Fig.4 shows a relation between the variances of the distribution of $\vec{v}$ and the dimension of $\vec{v}$. The horizontal axis is the dimension of $\vec{v}$ and the vertical axis is the variances of the distribution of $\vec{v}$. For normalization, values of vertical axis was divided by 1460, which is the maximum value of the payload length. (MTU of link we monitored is 1500 bytes.) We consider 100 flows for each application.

From Fig.4, we can see that the variances of each application increase in proportion as the dimension of $\vec{v}$. This means

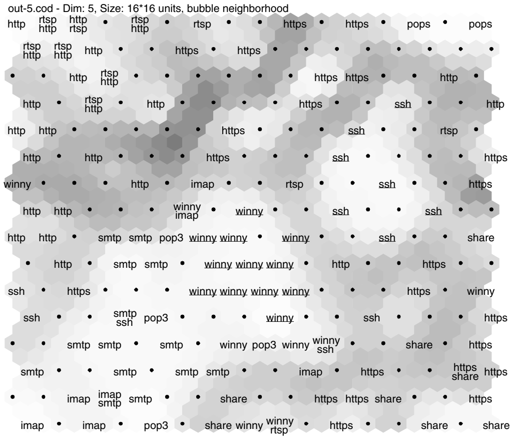

Fig. 2. Distribution of $\vec{v}(n{=}5)$

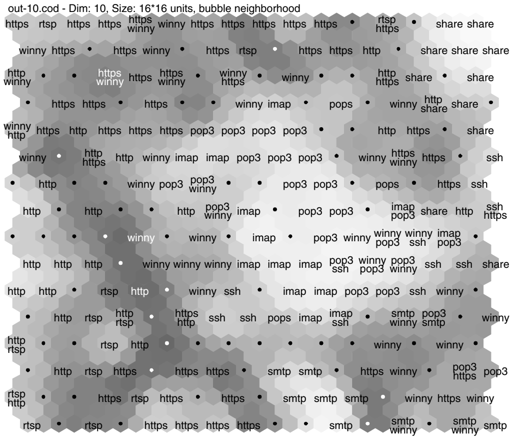

Fig. 3. Distribution of $\vec{v}(n{=}10)$

that some packets of early stage of a flow include similar messages, and they have similar sizes. After communication of similar messages, such as an authentication message, the variances of $\vec{v}$s are large because transmission of contents whose sizes are varied would begin.

These results justify our assumption that the lengths of the first few packets of flows of the same application are similar. In addition, the results show that we should not select dimension of $\vec{v}$ too large to identify applications accurately because the similarity among flows of the same application becomes low as the number of considered packets increase. However, if there are applications which has similar transition patterns in the first few packets of their flows, it is necessary for identification methods to distinguish them by using more packets. Therefore, we need an adequate way to increase dimension without decrease in similarity.

C. Transition pattern expression using inverse of payload length

A reason of a similarity decrease is transmission of contents such as the body message of email and HTML files. Because the size of contents range widely, the transition pattern of the payload lengths of contents communication does not exhibit large similarity.

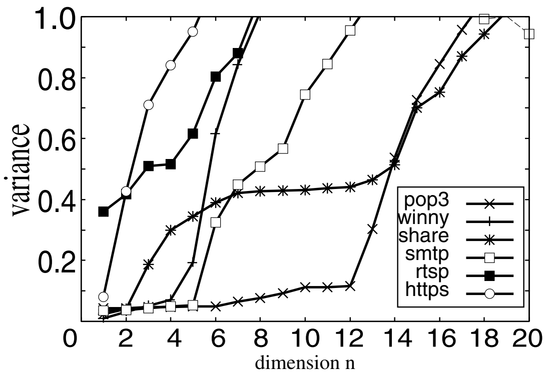

Fig. 4. Variances of $\vec{v}$

This problem can be solved by adopting appropriate dimension for each application. However, appropriate dimensions are different for each application. It is difficult to set up the appropriate dimensions in advance, and a new analysis to determine the dimension that would be needed if a new application would be begun to utilize. Thus, another approach is necessary to solve the problem of the vector dimension.

In this section, we propose to use a vector $\vec{v}_{inv}$ whose elements are inverses of payload length. $\vec{v}_{inv}$ is defined as

$$\vec{v}_{inv} = [\frac{1}{v_1}, \frac{1}{v_2}, ..., \frac{1}{v_n}]^T. \tag{2}$$

Using the inverse of payload length, the difference of the payload lengths of typical interaction between different application flows will be large because the payload lengths of the typical interaction are generally small. And the difference of the payload lengths of contents communication will be small because the size of packets of contents communication change around MTU. Consequently, the inverse of payload length has an effect on emphasizing the difference among the payload length of the typical interaction of different applications and defuse the large variance of the payload length of contents communication.

We evaluate the distribution and the variances of $\vec{v}_{inv}$ the same way in Sec.III-B. Fig. 5 is the SOM output when the dimension of $\vec{v}_{inv}$ is 10. Fig.5 shows that clusters of same applications are kept when dimension of $\vec{v}_{inv}$ is increased.

Fig.6 is the variances of $\vec{v}_{inv}$. Fig.6 shows that in all applications except Share, variances of $\vec{v}_{inv}$ is smaller than that of $\vec{v}$. The reason of large variance of Share is that Share has more than two different transition patterns.

D. Classification Method

In this section, we proposed a method to discriminate applications by using transition pattern of payload length. Our proposed method is consisted of two steps: learning phase and identification phase.

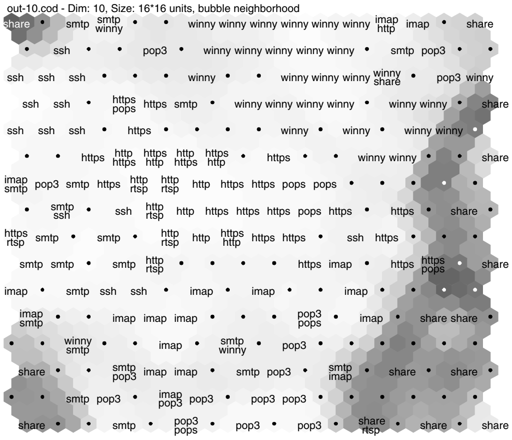

Fig. 5. Distribution of $\vec{v}_{inv}$(n=10)

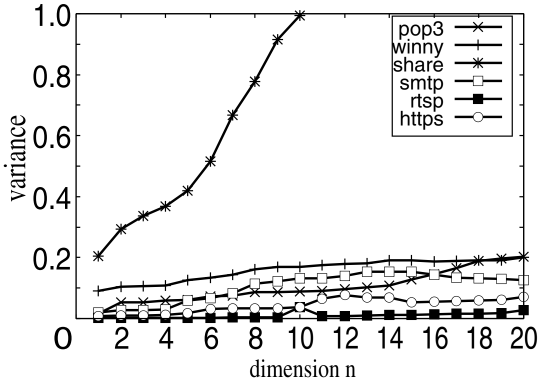

Fig. 6. Variances of $\vec{v}_{inv}$

In the learning phase, our method generates codebook vectors, which represent the distributions of $\vec{v}$ of each application, using labeled sample flows which are identified their applications in advance. In the identification phase, our proposed method compares distances between a vector which is extracted from a newly observed flow and codebook vectors, and determines that the flow is transmitted by the application of the codebook vector nearest to its flow.

IV. Evaluation

In this section, we evaluate the proposed method by identifying real network traffic. We collected 3000 flows of 10 applications (300 flows per an application) for learning. Test data set consists of flows shown in Table I. Two algorithms to generate codebook vectors are applied, which are Optimized Learning Vector Quantization and K-Means.

A. Identification Accuracy using Optimized LVQ1

We calculated 200 codebook vectors by Optimized Learning Vector Quantization 1 (oLVQ1) [11]. LVQ is an algorithm which is similar to SOM and modified for supervised learning algorithm. oLVQ1 is a variation of LVQ, which is fast and accurate. We used LVQ_PAK [12] to create codebook vectors.

TABLE I
Number of test samples

Application	Number of Flow
http	2403
smtp	2349
pop3	4633
imap	1184
ssh	1039
winny	2913
share	1292
https	1898
pops	2733
rtsp	554
Total	20998

Accuracy of each application when dimension n is six is shown in Table II. An identification accuracy for application X is defined as

$$accuracy = \frac{Nt_X}{Nf_X}. \tag{3}$$

We define the number of true positives of application X(Nt_X) as the number of correctly identified flows of application X, and the total number of flows of application X(Nf_X).

TABLE II
Classification accuracy using oLVQ1 (n=6)

	accuracy [%]	
application	$\vec{v}$	$\vec{v}_{inv}$
http	73.5	51.1
smtp	98.5	99.5
pop3	100.0	98.6
imap	90.2	97.3
ssh	96.5	98.0
winny	87.4	94.4
share	98.0	96.3
https	66.2	82.5
pops	99.6	99.6
rtsp	59.5	88.1
Total	92.4	93.3

Table II shows that most applications are identified correctly by using the first six packets of a flow.

Identification accuracy for http, https and rtsp are worse than others. The reason is that typical interaction of these applications consist of small number of packets. Thus, contents communication has already started in the first six packets of the flow. Another reason is that behavior of http and rtsp flow is similar [17].

Dimension of vectors versus identification accuracy of Winny, which is the most popular P2P file sharing software in Japan, is shown in Fig.7. The horizontal axis is the dimension of $\vec{v}$ or $\vec{v}_{inv}$. The vertical axis is accuracy for Winny, which is defined as

$$R_{winny} = \frac{Nt_{winny}}{Nf_{winny}}. \tag{4}$$

Fig.7 shows that the identification accuracy decreased when dimension increased by using $\vec{v}$. But by using $\vec{v}_{inv}$, accuracy did not decrease. This means that $\vec{v}_{inv}$ reduces the adverse effect of contents communication. This result means that it is effective to use $\vec{v}_{inv}$ when the dimension need to be increased.

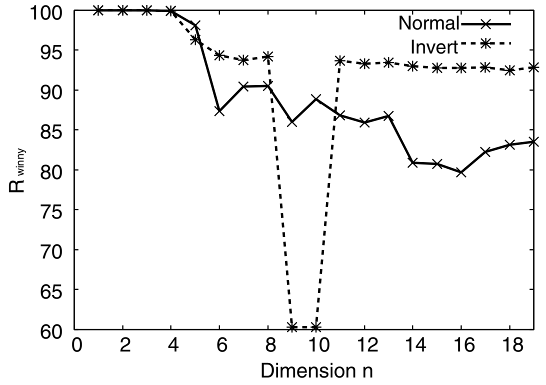

Fig. 7. Accuracy of Winny

When the dimension $\vec{v}_{inv}$ is nine and ten, the accuracy was down.

Why these decreasing was occur is unclear. But, most of them are misidentified only in the case that the dimension is nine or ten. When dimension is more than 11, flows which were misidentified in these two dimensions are identified correctly. Thus, the reason is that payload length of 9th or 10th packet effect badly for identification and 11th packet can mask their bad effect.

A simple way to identify Winny flows properly is classifying with other dimension. In fact, when we select other dimension, they were classified properly.

B. Identification Accuracy using K-means

As another generating method of codebook vectors, we applied K-Means clustering algorithm, which is one of the quickest and the simplest clustering algorithm. Accuracies of each application when dimension n is six are shown in Table III.

TABLE III
CLASSIFICATION ACCURACY USING K-MEANS (n=6)

	accuracy [%]	
application	$\vec{v}$	$\vec{v}_{inv}$
http	54.9	62.7
smtp	99.1	99.4
pop3	99.8	99.7
imap	99.2	98.4
ssh	85.6	86.3
winny	86.2	94.5
share	98.0	98.5
https	64.9	90.6
pops	99.6	99.6
rtsp	95.6	58.0
Total	92.3	94.4

Table III shows that transition pattern of payload length is effective without depending clustering algorithm. The flows of http and rtsp were also misidentified. Https flows were classified correctly by using $\vec{v}_{inv}$. However, http and rtsp were not. These two applications should be classified with other method.

V. CONCLUSION

In this paper, we proposed an identification technique of network applications based on transition pattern of payload length of packets without using port numbers. We assume that applications has typical interactions in the beginning of flows. We confirmed that flows of an application show similarity in their transition pattern of payload length in the beginning of flows.

However, we found a problem which caused by increasing the number of considering packets. When the number of packets increase, the similarity of the transition pattern of packet lengths will decrease. This problem due to an adverse effect of contents communication, such as the communication of body messages of email and HTML files, and so on. The payload length of packets from the contents communication tends to be large. Thus, we proposed to use the inverse of payload length. By using the inverse of payload length, packets of contents communication is weighed lightly, and packets of the typical interactions which are relatively small is weighed heavily.

Through experiment, we show that the inverse of payload length is more effective than the normal payload length when the number of considering packets increased. However, some applications were misidentified, such as http and rtsp. To identify these application correctly, other approaches, such as using payload information of packets are needed. And our proposed method needs codebook vectors which represent the distribution of each application in advance. Consequently, our proposed method cannot identify “Unknown” applications. This is our future effort.

REFERENCES

[1] J. N. S. Association. Information incident servey report. http://www.jnsa.org/resullt/2005/20060803_pol01/index.html.

[2] L. Bernaille, R.Teixeira, and K. Salamatian. Early application identification. In *In Proc. of Confernce on Future NetworkingTechnologies*, Dec. 2006.

[3] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. In *ACM SIGCOMM Computre Communication Review*, 2006.

[4] F.Nakamura, T. Matuda, Y.Wakahara, and Y.Tanaka. Traffic feature analysis and application discrimination. In *IEICE technical report, NS2006-80:57-62*, Sep. 2006.

[5] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. Acas: Automated construction of application signatures. In *SIGCOMM '05 Workshops*, Augst 2005.

[6] J.Erman, M.Arlitt, and A.Mahaniti. Traffic classification using clustering algorithms. *MineNet '06: Proceedings of the 2006 ACM SIGCOMM workshop on Mining network data*, pages 281–286, 2006.

[7] T. Karagiannis, A.Broido, N.Brownlee, kc.claffy, and M. Faloutsons. Is p2p dyng or just hiding? In *In Globecom*, 2004.

[8] T. Karagiannis, A. Broido, M. Faloutos, and K.C.Claffy. Transport layer identification of p2p traffic. In *IMC'04*, October 2004.

[9] T. Kitamura, T. Shizuno, and T. Okabe. Application classification method based on flow behavior analysis. In *IEICE technical report, NS2005-136:13-16*, Dec. 2005.

[10] T. Kitamura, T. Shizuno, and T. Okabe. Traffic identification method with packet type transition pattern analysis. In *IEICE technical report, NS2006-28:29-32*, May. 2006.

[11] T. Kohonen. *self Organized MAP*. 1996.

[12] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. Lvq pak: A program package for the correct application of learning vector quantization algorithms, 1992. http://citeseer.ist.psu.edu/kohonen92lvq.html.

[13] A. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In *PAM '05*, 2005.

[14] A. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *SIGMETRICS'05*, 2005.

[15] M.Tai, S.ATA, and I.Oka. A classification method for bulk/real-time traffic based on flow statistics. In *IEICE Technical Report NS2006-28:29-32*, May. 2006.

[16] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 135–148. ACM Press, 2004.

[17] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (rtsp), 1998.

[18] S. Sen, O. Spatscheck, and D. Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 512–521, 2004.

[19] T.Karagiannis, K.Papagiannaki, and M. Faloutsos. Blinc: Multilevel traffic classification in the dark. pages 229–240, 2005.

[20] Y. Waizumi, A. Jamalipour, and Y. Nemoto. Network application identification based on transition pattern of packets. In *WRECOM*, Rome, Italy, 2008.

[21] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. In *ACM SIGCOMM Computre Communication Review, Vol.36,Number 5*, 2006.