Application Traffic Classification using Statistic Signature

Hyun-Min An[1], Myung-Sup Kim[1]
[1]Dept. of Computer and Information Science
Korea University
Korea
{queen26, tmskim}@korea.ac.kr

Jae-Hyun Ham[1,2]
[2]The 2nd R&D Institute – 1
Agency for Defense Development
Korea
jaehyun_ham@korea.ac.kr

Abstract— **Networks today are becoming more complex and diverse because of the appearance of new applications and services. The importance, therefore, of application-level traffic classification is increasing daily. Application-level traffic classification has become a very popular area of study. Although many proposals have been presented, including port-based, payload-based and machine learning-based methods, the method that can manage all traffic has not yet been developed. More recently, methods based on statistical flow information have been studied. In this paper, we propose an application-level traffic classification methodology using the statistic signature. Our method creates a statistic signature using payload size, transmission order, and direction of the first N packets in the flow, and uses this to classify application traffic. Then, using a verification system, we prove the feasibility of our method and show its high accuracy.**

Keywords- Traffic classification; Statistic signature, Application Traffic

I. INTRODUCTION

Networks today are becoming more complex and diverse because of the appearance of new applications and services. For effective operation and network management, traffic classification has become mandatory. It is expected that the reliance on networks will increase in the future. The importance of traffic analysis will grow with this need [1]. Campus and enterprise networks provide policies such as QoS and SLA for the efficient management and operation of network resources. For example, in schools and public institutions, there are policies to control traffic that consume excessive network resources and are not related to the organization, such as P2P and game traffic. For these policies, fast and accurate traffic classification in the application layer is essential [1,2,4].

Application-level traffic classification is a process that collects network packets and determines the identity of the application [1,3]. Accurate real-time application-level traffic classification is an important part in determining the reliability of monitoring and controlling application traffic for each application (including cost per application, traffic control per application, CRM, SLA, application layer security).

Because of the importance of application-level traffic classification, many studies have recently been presented. Existing classification methods include port-based [2], payload signature-based [3], machine-learning-based [10], and flow correlation-based [1,4]. However, because traffic is changing due to the updating of applications or emergence of new applications, application-level traffic classification has become more difficult. Therefore, additional research is required for more accurate traffic classification.

Today, there are traffic classification studies [7] that use statistical flow information to overcome the drawbacks of the conventional methods and to classify application traffic more quickly and accurately. A classification method that uses statistical information analyzes the traffic using machine-learning algorithms and statistical characteristics such as the window size, packet size, and inter-packet arrival time. This method has advantages. It works for encrypted traffic, the usage of which is increasing recently. In addition, it does not need to analyze the payload information of the packets and so it can classify the traffic quickly. The drawback with this approach is that it has to wait until the end of the flow to complete the statistical information. To overcome this problem, studies [5] that use the first N packets of a flow have been implemented. Their results are not detailed, however, because they classify the traffic by application protocol units such as FTP, HTTP/HTTPS, and POP3.

In this paper, we propose an application-level traffic classification method that creates a statistic signature for each application using payload size, transmission order, and direction of the first N packets in the flow, and then uses this to classify the application traffic. Our proposed method has three main advantages.

First, it is very accurate. It is not possible to classify all application traffic using only the direction and size of the payload, but it classifies the traffic correctly. In the field of network traffic control, incorrect traffic classification can be a bigger problem than unclassified traffic. This method can be utilized in these areas.

Second, it classifies the traffic based on the application process, which is more detailed than the application protocol. Proposals that classify traffic using the direction and size of the packet have been presented before, but their method classifies the traffic by application protocol, producing results that are

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (2010-0020728) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2007483)

* Corresponding Author : Myung-Sup Kim (tmskim@korea.ac.kr)

not sufficiently detailed. The proposed method classifies the application traffic, used in a real network, by the application process to provide more detail for network management policies such as QoS and SLA.

Third, this method is suitable for real-time traffic classification. It can quickly classify the traffic because it uses only the direction and size of the first N payload packets of the flow, and does not analyze the payload information.

The remainder of this paper is organized as follows. Related studies are briefly reviewed in Section II. In Section III, we present the details of the algorithm for the proposed method. The evaluation of the proposed method using a verification system is discussed in Section IV. We present the conclusions and future work in Section V.

II. RELATED WORK

In today's Internet environment, there are many applications making application-level traffic classification difficult. In the past, applications that accounted for most of the Internet traffic, including HTTP, telnet, e-mail, FTP, and SMTP, had port numbers under 1024. Traffic classification based on port information from the IANA definition, therefore, was adequate to achieve reliable and accurate results. However, today, a number of new applications have emerged, such as passive FTP and streaming applications. These use multiple sessions for multiple and simultaneous communication, and port numbers can be dynamically selected. Port-based analysis, therefore, is no longer effective. To overcome these drawbacks, methods of traffic classification obtaining the dynamically generated port information by referring to the contents of the application protocol were introduced in mmdump [8] and SM-MON [9]. The advantage of these methods is high classification reliability because they can determine the exact port information by referring to the contents of the protocol. However, these methods can only be used for application traffic where the application protocol is known to the public, such as RTSP, MMS, and SIP. It cannot be applied to all Internet traffic. Many applications, including P2P programs, which consume significant traffic on the current Internet, do not open the port and application protocol information. Therefore, a method for extracting a dynamic port number by referencing the contents of the application protocol cannot be used in most cases.

In order to resolve these difficulties, several traffic classification methods have been proposed. Traditional methods can be divided into three types: signature-based [3], traffic-correlation-based [1,4], and machine-learning-based [5,10].

First, signature-based classification methods analyze the traffic that is generated in a particular application to extract features called signatures. These can be used to distinguish them from other applications. They classify the traffic using these signatures. These methods are usually very accurate. Because signatures are usually extracted by hand, however, they cannot properly respond to application change. Furthermore, they cannot classify applications when it is difficult to extract signatures. For example, payload-signature-based traffic classification methods cannot be used for encrypted traffic.

Second, traffic-correlation-based classification methods use relational information in traffic flows to classify traffic. They use features such as address system (IP address, port number, protocol), occurrence time, and occurrence form of the traffic. These methods can classify traffic using many application characteristics. However, there is no clear algorithm for the use of these characteristics and so these methods look for a threshold that indicates a percentage of the optimal analysis using trial and error. Therefore, when applied in the actual Internet environment, these methods do not provide adequate reliability.

Third, machine-learning-based classification methods use classification and clustering techniques of machine-learning to classify the traffic. They use items that can be features of Internet traffic (port number, flow duration, inter-packet arrival time, packet size). These methods use high quality machine-learning techniques to classify traffic. However, when this method is applied in the Internet environment, the accuracy of the classification will decrease because they only collect and classify traffic in a limited range. Additionally, in the case where the methods do have a high analysis rate, the traffic is classified into applications that are trained. They are not suited to handle new applications.

In this paper, we propose a statistic signature-based traffic classification method that belongs to the signature-based group. There are other studies that use direction and size of packets, as this paper does, but they classify traffic by application protocol, and so their results are difficult to apply in different fields. In this paper, we provide a classification method to classify the traffic into application processes. Our method is more accurate, faster and easier than solutions using a machine-learning algorithm. We applied our method on an actual campus network to verify the performance and validity.

III. APPLICATION TRAFFIC CLASSIFICATION USING STATISTIC SIGNATURE

It is necessary to determine the criteria for classifying traffic before actually performing the classification task. Many related works classify the traffic by application protocol. Our method classifies the traffic according to the application process. Traffic classification by application process provides a more detailed analysis result than classification according to the application protocol, making it more useful to more fields.

A. Statistic signature

Statistic signature is a unique characteristic that can distinguish one application from another using statistical information that is obtained from the packets in a flow (for example, packet size, window size, capture time).

In this paper, we use statistical information consisting of payload size, direction of packets, and order of packets in a

flow. The payload size of a packet is the payload length in a packet that has payload data. The direction is expressed as two values, '+' or '-'. In the case of TCP, a '+' indicates that the packet transfers from the client to the server, and '–' indicates that the packet transfers from the server to the client. Because the distinction between the server and the client is not clear in UDP, the meaning of '+' / '-' only indicates that the directions are opposite. We start by denoting the first packet's direction as '+', and then determine the direction of the following packets by comparing them with first packet's direction. The same direction is '+', otherwise it is '-'.

B. Packet Size Distribution

In general, the first packets in a flow consist of the predefined rules of the application. Accordingly, the payload size distribution of the first N packets in the flow of an application can differ from other applications.

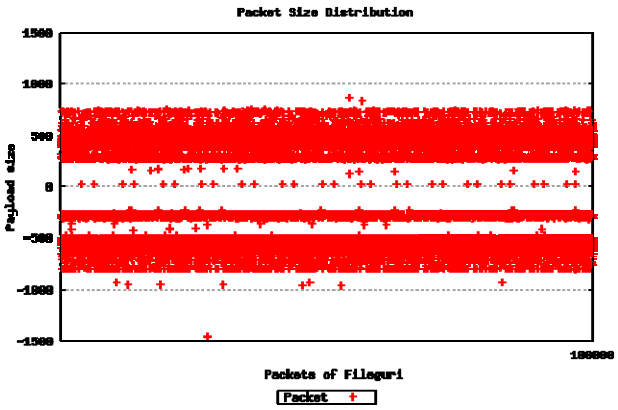

Figure 1. Packet size distribution of fileguri

Figure 1 shows the payload size distribution for 100,000 payload data packets for an application named "Fileguri" that is popularly used on the campus network. It is a web hard service. The x-axis represents the packet number in capture time order. The y-axis represents the direction and size of the packets, The size of a packet is in the range -1460 to +1460. Most of the packets have a particular payload size, such as +250 ~ +600, -300, -500 ~ -700. This shows that each application does have a particular payload size that is commonly used. However, it does not mean that all applications have payload sizes similar to "Fileguri". There are many applications using HTTP, such as Internet Explorer. An HTTP flow has a variable payload size, so the distribution of the payload size cannot completely classify the traffic. In this paper, we classify all application traffic, except HTTP.

C. Extract Statistic Signature

Figure 2 shows a flow chart of the proposed method. It is divided into two primary functions: signature generation and traffic identification. The signature generation section first converts the flows into packet size distribution vectors and divides the flows by process. The process flows then enter the flow grouping step and are grouped by the distance similarity of the flow vectors. Then, the groups of all the processes are optimized and the signature for each group is generated. The traffic identification portion generates a flow using packets that are continuously collected. It then converts the flow into a vector and matches the flow vector with the signature to identify the corresponding application name.

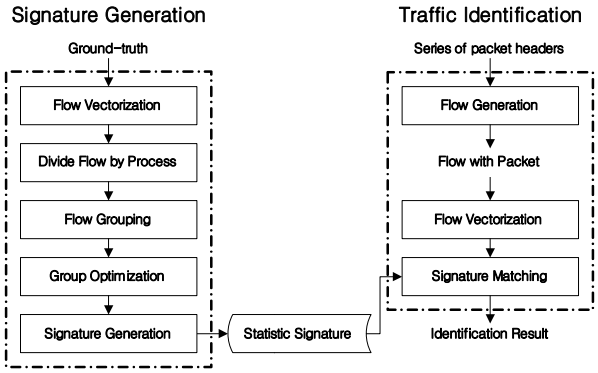

Figure 2. Flow Chart of Application Traffic Classification System Using Statistic signature

Training Traffic Traces

There are certain requirements for training traffic traces to generate statistic signatures. First, the traces must have traffic from all the targeted processes. Second, the traces must have more than a minimum number of flows from each process. A signature generated by too little traffic has low reliability and can reduce the traffic that can be classified. Third, the traces should be collected from multiple hosts. If the traffic of a process is collected from only one host, the features could be influenced by the characteristics of that host. For example, many applications have a function that can change their port number. The user changed port number can be used as a signature for that host, but it cannot be used as a signature for other hosts.

In this paper, we collected the target process traffic from our campus network. The number of flows of each process was greater than 1,000 and the flows were chosen randomly.

Flow Vectorization

In the previous section, we showed the possibility of classifying traffic using the payload size distribution of the packets. To use this for generating the signatures and classifying the traffic, we must decide how to express the direction and size of the payload of first N packets in a flow. We use vector representation for this purpose.

We assume that f_k is the k -th input flow, and $s_{k,1}$ is the payload size of the i -th packet of the k -th flow. $s_{k,1}$ can have a direction value '+' or '-'. v is a function that expresses a flow into an N-dimensional vector. It can be written as shown in (1).

$$v(f_k) = \{s_{k,1}, s_{k,2}, ..., s_{k,N}\} \quad (1)$$

For flow grouping and traffic classification, we need a measurement of the distance between two vectors. In our method, we use the distance per dimension. The distance between two flow vectors is also a represented vector. The i-th element of the distance vector, d, is the difference between the i-th element of the two flow vectors. It can be expressed as shown in (2).

$$d(v(f_k), v(f_j)) = \{|s_{k,1} - s_{j,1}|, |s_{k,2} - s_{j,2}|, ..., |s_{k,N} - s_{j,N}|\} \quad (2)$$

Flow Grouping

Each flow group has five attributes as shown in Table 1. The Process code is a positive integer that represents each application process. It is assigned incrementally and exclusively. The L4 protocol is the protocol of the transport layer that is used by the application. It has a value such as UDP or TCP. The Centroid vector represents the center point of the vectors in a group. The i-th element of a Centroid vector is the average size of the i-th element of all the flows in the group. If $C_i(G)$ is the i-th element of the Centroid vector of group G, it can be derived using equation (3).

$$C_i(G) = \frac{1}{m} \times \sum_{j=1}^{m} s_{j,i}, \ f_j \in G, \ m = \#\ of\ Flow \quad (3)$$

The Dimension indicates the number of elements in the Centroid vector. It is the number of packets in the flows belonging to the group, so it can takes any value from 1 to N. In this paper, we assign N = 5.

The DT-vector (Distance Threshold Vector) indicates the distance threshold of each dimension of the group using a vector expression. It decides the size of each group. If $dt_i(G)$ is the i-th element of the DT-vector, then $R_i(G)$, the range of the i-th element of the group, can be expressed as shown in (4).

$$R_i(G) = \{x \mid C_i(G) - dt_i(G) \le x \le C_i(G) + dt_i(G)\} \quad (4)$$

When all of the elements in a flow vector satisfy the value of $R_i(G)$, the flow will be grouped in G. We state that the flow is included in the group G, and that G becomes the including Group. The initial value of the DT-vector is set to all 10's.

Table 1. Attributes of a Group

Attribute	Example
Process code	101001
L4 protocol	UDP
Centroid vector	{+20,+30,-50,+20,-30}
Dimension	5
DT-vector	{10, 10, 10, 10, 10}

In the flow grouping step, we group the flows of a process into several groups having different group attributes. The grouping proceeds with the flows that occurred in the same process. Figure 3 shows the pseudo-code of the grouping algorithm. All flows are converted into vectors and divided by process to be input to the flow grouping step. We process the flows individually. We search the existing groups to find a group that could include the flow. If there is group that can include the flow, the flow is grouped into that group and the Centroid vector of that group is re-calculated. If there is no group that could include the flow, a new group is created and the attributes set by referring to the flow.

Flow Vectorization

divide flow by process

```
1: procedure Flow Grouping(flow)
2: search includingGroup in all groups in process
3: if includingGroup found {
4:     insert the flow into includingGroup
5:     recalculate centroid vector of includingGroup
6: }
7: else
8:     make new group
9: end procedure
```

Figure 3. Pseudo-code for grouping algorithm

Group Optimization and Signature Generation

The group optimization step is organized into two courses. The first is the course to eliminate any outliers and the second is to minimize the DT-Vector.

In the first course, we eliminate the flows that fall outside of their group. Then, we eliminate the groups that have too few flows. The purpose of the second course is to minimize the range of the groups. Figure 4 shows an example of group optimization in a 2-dimensional Euclidean space.

The range of the group changes continuously during the grouping process, and so there are some flows that fall out of their groups, as shown in Figure 4, left. These flows will not belong to the group eventually, so they are eliminated from the group to reduce misclassification. Figure 4, right, shows the results after removing all of the outliers, and the minimization of the elements of the DT-Vector based on the farthest flow, in element units, from the Centroid vector.

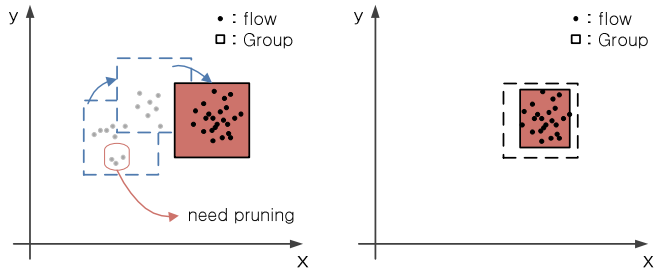

Figure 4. Groups Optimization

After the group optimization step, the signature generation step is initiated. The signatures are extracted from the groups, one for each group. The signature has six attributes as shown in Table 2. The attributes have the same value as the attributes of the group, except the Representative vector of the signature replaces the Centroid vector of group. The smallest value of each element in the flow that belongs to the group is an element of the Representative vector. This is used to reduce the calculation time in the traffic classification. For classifying the traffic, we must check whether the flow belongs to the range of the signature or not. The Centroid vector is a median value of the group, so it has +/- forms of the threshold value. If we wanted to use it, we would need two numerical computations and two condition checks to measure the flow against a signature. However, the Representative vector is the minimal value of the group, so it has only a '+' threshold value. Therefore, we need just one numerical computation and one condition check when we use it. In other words, in the worst case, using the Representative vector rather than the Centroid vector requires only half the computing cost. With today's Internet traffic, in real-time traffic classification, the performance difference between the two methods cannot be ignored. The Fixed Port, the additional feature of the signature, will have a value when all the flows in the group have the same server port number.

Table 2. Attributes of Signature

Attribute	Example
Process Code	1010001
L4 protocol	UDP
Representative vector	{+20,+30,-50,+20,-30}
Dimension	5
DT-Vector	{5, 10, 10, 5, 5}
Fixed Port	5004

Unlike in TCP, in UDP the server and client are not clear and so we assume that the host who issued the first packet is the client host. We use the property in UDP, as in TCP, that the client requests to the server first.

Traffic Classification Method

In the real-time traffic classification step, we collect and classify the traffic in the actual network. First, we generate the flows using incoming packets. We use the ordered set of packets from two end hosts that have the same 5-tuple information (source IP, destination IP, source port, destination port, protocol) for flow. The flow is converted to a vector and moves to the Signature Matching step. If a flow can be matched by the signature to one application, we make the decision that this flow is from that application. However, if the flow is matched by the signature to two or more applications, we define this as a signature conflict and we do not make a decision as to where the flow is from. We use the direction and size of the packet payload up to 5 to classify the traffic. Our method, therefore, is suitable for real-time classification.

IV. EXPRERIMENT RESULT

A. Traffic Trace

For verification, this paper selected the traffic between the Internet and the Korea University campus network. The characteristics used for the traffic classification included the direction of the packets, so we had to collect the traffic in both directions. The campus network was configured with one router. We had to collect the traffic in the router, therefore, using mirroring. The training data set for the signature generation consisted of the flows and the payload packets (the packets had payload data) that belonged to the flow, and the traffic was collected based on the process. Because the purpose of this method was to classify the traffic by application process, we excluded the TCP control packets that appeared in the transfer layer. These included SYN, ACK without data and Keep-Alive packet.

It was very important for the training data set to accurately collect the traffic by each process. In this paper, we installed TMA (Traffic Measurement Agent) [6] to some end hosts in the campus network to establish the ground-truth. This agent was able to guarantee higher reliability than the using the results of the particular classification method [4].

B. Evaluation Element

We used completeness and accuracy as evaluation elements. Completeness is a measurement of how much traffic was classified. Accuracy is a measurement of the accuracy of the traffic that was classified. It is determined by comparing the classification results with the ground-truth. It is expressed as the rate of accurately classified traffic per total classified traffic. The accuracy is divided by the overall accuracy and accuracy per application to derive the precision and recall for each application. These evaluation elements are expressed by flow, packet, and byte unit to provide more detailed information.

C. Validation Result

Table 3 shows that the completeness and overall accuracy by flow, packet and byte unit. We can see the high accuracy compared to the low completeness. There are two reasons for this. The completeness decreases because there are applications that are part of the traffic but are not used in the signature generation process. In addition, in the traffic identification process, if a flow has been matched with signatures from more than one process, this flow will have no matching result for eliminating misclassification.

Table 3. Completeness and Overall Accuracy

	Completeness	Overall Accuracy
Flow	57.48%	99.69%
Packet	47.71%	99.97%
Byte	53.70%	99.99%

Table 4 shows that the precision and recall by process. The precision for every process except Skype and Kartrider is greater than 99.9%. The reason for the lower precision of Skype and Kartrider is that there are many flows in these two applications that consist of only one or two packets and so they are easily classified by other application signatures. Flows with only a few packets have a low feature vector dimension. Consequently, the range of the application signature is easily overlapped by others. In summary, there was high precision in the classification results. We can, therefore, make the statement that the results support the reliability of the proposed method.

Table 4. Precision and recall by application

Application	Precision	Recall
Dropbox	99.99%	61.97%
uTorrent	99.97%	55.17%
Nateon	99.99%	33.66%
Skype	97.90%	78.10%
Kartrider	97.60%	24.59%

The recall values for each application have varied values, but they are consistently low. This means that the feature representing the direction and size of the payload packets cannot cover all the traffic of a process and so it is not possible to generate a signature for all the actions of the traffic. Therefore, another signature model or feature that can classify this unclassified traffic is needed. We have left this problem for a future work.

The traditional traffic classification method based on payload signature is widely used because of its high accuracy and completeness. The proposed method has a similar degree of accuracy. Because it does not analyze the payload data, however, it can classify the traffic in a shorter time. In addition, it has a more detailed result than the traditional method using similar features [5] because it classifies the traffic by process. This means it can be more widely utilized.

V. Conclusion and Future Work

In this paper, we proposed a method that extracts a statistic signature from the transfer order, direction and payload size of the packets in a flow and uses this to classify the traffic. Our method can classify application traffic with high accuracy. Compared to traditional studies that use similar features, our method has a more detailed unit for classification criteria, and so is more efficient for many network management policies including QoS and SLA. It does not analyze the payload data but uses only payload size and direction of first N packets in the flow. Consequently, it can classify traffic quickly and it is suitable for a real-time traffic classification environment.

However, there are several problems including the processing of unclassified traffic. In a future work, we plan to research a method that can classify Internet traffic with high accuracy and completeness using statistic signature. We also plan to research the various factors that can influence the traffic classification method using statistical information.

References

[1] Myung-Sup Kim, Young J. Won, and James Won-Ki Hong, "Application-Level Traffic Monitoring and an Analysis on IP Networks", ETRI Journal, Vol.27, No.1, pp.22-42, Feb., 2005.

[2] IANA port number list, IANA, http://www.iana.org/ assignments/port-numbers.

[3] Liu, Hui Feng, Wenfeng Huang, Yongfeng Li, Xing "Accurate Traffic Classification", Networking, Architecture, and Storage, 2007. NAS 2007. International Conference

[4] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. "BLINC: Multilevel Traffic Classification in the Dark," Proc. of SIGCOMM 2005, Philadelphia, PA, Aug., 22-26, 2005.

[5] Bernaille, L., Teixeira, R., Salamatian, K.: Early application identification. In: CoNext 2006. Conference on Future Networking Technologies., 2006.

[6] 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, pp.160-167, April, 7-11, 2008.

[7] Rentao Gu, Minhuo Hong, Hongxiang Wang, and Yuefeng Ji, "Fast Traffic Classification in High Speed Networks" Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2008, LNCS 5297, Beijing, China, pp.429-432, Oct., 22-24, 2008.

[8] Jacobus van der Merwe, Ramon Caceres, Yang-hua Chu, and Cormac Sreenan "mmdump - A Tool for Monitoring Internet Multimedia Traffic," ACM Computer Communication Review, 30(4), October, 2000.

[9] Hun-Jeong Kang, Myung-Sup Kim, and James Won-Ki Hong, "Streaming Media and Multimedia Conferencing Traffic Analysis Using Payload Examination," ETRI Journal, Vol.26, No.3, pp.203- 217, Jun., 2004.

[10] T. T. Nguyen and G. Armitage. "A survey of techniques for Internet traffic classification using machine learning." IEEE Communications Surveys and Tutorials, to appear, 2008.