Real-time Traffic Classification Algorithm Based on Hybrid of Signature Statistical and Port to Identify Internet Applications

Hamza Awad Hamza Ibrahim 1
hamysra76@hotmail.com
Mohamed Saad Mahoub 1
masnmas2@yahoo.com
1 Universiti Teknologi Malaysia
Faculty of Electrical Engineering

Sulaiman Mohd Nor 1
sulaiman@fke.utm.my
Haitham A. Jamil 1
haithamjamil@yahoo.com

Izzeldin Ibrahim Abdelaziz 2
izzeldin@ump.edu.my
Haitham A. Jamil 2
Universiti Malaysia Pahang
Faculty of Electrical and Electronics Engineering

Abstract—Internet traffic classification gained significant attention in the last few years. Most of the current classification methods were only valid for offline classification. The three common classification methods i.e. port, payload and statistics based have some limitations. This paper exploits the advantages of all the three methods by combining them to produce a new classification algorithm called SSPC (Signature Statistical Port Classifier). In the proposed algorithm, each of the three classifiers will individually classify the same traffic flow. Based on certain priority rules, SSPC makes classification decisions for each flow. The SSPC algorithm was used to classifying four types of Internet applications in two stages, initially offline and later online. The results of both cases show that SSPC is the higher accuracy when compared with other classifiers. In addition, as demonstrated in the real time online experiments done, SSPC algorithm uses a short time to classify traffic and thus it is suitable to be used for online classification.

Keywords: Internet traffic Classification; Machine Learning; hybrid classifier; port classification; signature classification

I. INTRODUCTION

Internet service providers (ISPs) and network operators are mostly interested in knowing the amount of traffic carried by their networks for the purposes of optimizing network performance and security issues. Therefore, Internet traffic classification is something valuable, particularly for interactive traffic applications such as VoIP and online games.

Simple classification assumes that most applications use well-known port numbers, and the classifier uses this port number to identify the application type. However, most Internet applications use unknown port numbers, or more than one application uses the same port number, which indicates the failure of port base classification [1]. Another classification method is payload based (deep packet inspection), which is individual packet inspection, looking for unique signatures. However, using this technique faces two problems; first, it is difficult to detect non-standard ports by using packet inspection because these packets are encrypted. Second, deep packet inspection touches on users’ privacy. In order to solve the problem of past classification methods (base port and payload inspection), machine learning (ML) technique was developed. ML [2] [3] uses artificial intelligence to classify IP traffic, which provides a powerful solution by extracting the right information from application features [4]. Moreover, some of the ML algorithms are suitable for Internet traffic flow classification at a high speed [5]. Most of the proposed ML classification methods are limited to offline traffic classification and cannot support online classification [6]. Online classification means, the decision of which packet belongs to which flow, assuming to be on the traffic speed. Such, like any hardware classifier (PacketShaper, SANGFOR), is installed on the network path to classify with the passage of the traffic.

The main problem that meets the online classification decision is the high speed of Internet traffic. It is difficult to take an online classification decision with huge amounts of Internet traffic. The question is how to divide the Internet traffic into flows, calculate flow patterns, and make classification decisions online with high Internet traffic speed. Most previous literature [6] [7] [8] [9] [10] [11] [12] [13] worked with classifiers using real time traffic, however only few of them [14] provided a classifier which can make an online decision.

This paper describes the development of an online Signature Statistic Port Classifier (SSPC) algorithm, which can identify Internet traffic shortly after traffic is captured. The task is also to classify network applications that use any TCP or UDP protocols. The classifier differs from others since it takes the classification decision based on three different parallel hybrid methods.

Section 2 describes and analyses the related works. General concepts of classification mechanisms, the three partial algorithms, and SSPC algorithm are dis-cussed in Section 3. In Section 4, the experiments were illustrated, and the results were shown. Finally, Section 5 provides the conclusion and limitations of the mechanism.
II. RELATED WORKS

The authors of [7], focus on traffic measurement in high-speed networks. The paper analyzes Internet applications to see the traffic measurement characteristics. Then design flow measurement system of high-speed networks based on Linux kernel is described. The system was built over some methods; firstly, from a perspective of Network Interface Card (NIC); the system designed a Hash function (group of rules) to classify packet processed by 32-bit systems instead of interrupt to communicate with OS. Second, the system identifies the new P2P service by calculating key hash value if there are no existing matching rules. Some shortcoming was observed such as the authors do not provide details what features are used to build hash rules. As well, the system defines any new traffic flow as new P2P. Moreover, the paper relies heavily on port numbers to identify traditional applications.

The paper [8] proposed a dynamic online method to classify Internet traffic. The method used two levels: overall traffic level and application level. Data mining algorithms are used to continue updated considered datasets. The proposed method has three parts: i) Traffic model; which is: preparing the dataset, selecting the features, and updating the model for the case of new application. ii) Traffic classification; to classify traffic based on the gained features. iii) Change detection; which is run periodically to check if there is a new application. While the paper title includes the words “online traffic classification”, but there is no online classification. Also, no details about traffic features used for classification are provided.

Another study in [9] proposed an approach for online classification for TCP traffic based on the first n packets. The approach used information from the first n packets to decide that to which kind of application the flow is belong to. The authors used correlation-based feature selection (CFS) [20] to select optimum features. However, it is something untrusted to classify flow include thousands packets based on the first few packets. This is because the first packets in many flows can differ statistically from the rest of the packets. Moreover, the paper did not provide details how the online decision was taken.

In [10], a traffic classifier based on support vector machines (SVM) was presented. The dataset includes three traces collected from three different places. Based on statistical features, the classifier used the first ten packets to identify the flow. As in the previous paper, while the paper title contains the words “online classification”, there are no online decisions. In addition, In addition, how to classify flow includes a huge amount of packets based only on ten packets.

The researchers of [11] propose a wireless mesh network traffic classification using C4.5. Sub-flow with application behaviors was applied to solve the problem of how to select represented sub-flow. Based on the statistical features of the first n packets, the classifier clusters the flow to one of the defined applications. Similar to the previous studies, the traffic datasets were captured in real time; however, there are no online classifications (capture and classification at the same time).

Another study [12] develops a classifier which quickly identifies an application at any point of a flow’s lifetime. Thus, the ML classifier was trained by using sets of features calculated from multiple sub-flows at different points. The classifier recognizes the flow either way (forward or backward) by features swapped called synthetic sub-flow pairs (SSP). Assistance of clustering technique (ACT) as unsupervised clustering ML technique was used to automate the selection process. The problem is that different datasets from different dates (maybe different networks) were considered for ML. This is not consistent with the rule of similarity of training and testing datasets in network environments.

[13] is a satisfactory work which proposes a method suitable for identifying the association with TCP flows. This is based on total data length sent by client (ACK-Len ab) or server (ACK-Len ba) before it received ACK packets. The proposed method was verified by using an ML classifier (C4.5) to classify four types of Internet applications (WWW, FTP, EMAIL and P2P). In the same manner as other research, no online classification was approved, only real time traffic was considered.

Another study [14] presents a network processor (NP) classifier, which is based on the online hybrid traffic to identify P2P traffic. The classifier is based on two stages: hardware static characteristics and software Flexible Neural Tree (FNT) [21]. In the first stage, the hardware classifier (based on payload and port) filters P2P traffic. In the second stage, the software classifier (based on ML statistical features) is used as statistical diction maker. However, the classifier depends on hardware (NP), which is an additional cost.

The researchers in [6] present an approach for traffic classification based on Naïve Bayes classifier. Similar to [9], the classifier identifies the flow by observing the first n packets. For online classification, when new flow is coming, the features packet size and inter-arrival time are extracted. Based on features values, the ML classifier directs the flow into one of the considered applications (HTTP, POP3, POP3SSL, SMTP, and FTP). However, the approach is seen as offline work because there is no discussion about online classification.

In [10], a traffic classifier based on support vector machines (SVM) was presented. The dataset includes three traces collected from three different places. Based on statistical features, the classifier used the first ten packets to identify the flow. As in the previous paper, while the paper title contains the words “online classification”, there are no online decisions. In addition, In addition, how to classify flow includes a huge amount of packets based only on ten packets.

The researchers of [11] propose a wireless mesh network traffic classification using C4.5. Sub-flow with application behaviors was applied to solve the problem of how to select represented sub-flow. Based on the statistical features of the first n packets, the classifier clusters the flow to one of the defined applications. Similar to the previous studies, the traffic datasets were captured in real time; however, there are no online classifications (capture and classification at the same time).

Another study [12] develops a classifier which quickly identifies an application at any point of a flow’s lifetime. Thus, the ML classifier was trained by using sets of features calculated from multiple sub-flows at different points. The classifier recognizes the flow either way (forward or backward) by features swapped called synthetic sub-flow pairs (SSP). Assistance of clustering technique (ACT) as unsupervised clustering ML technique was used to automate the selection process. The problem is that different datasets from different dates (maybe different networks) were considered for ML. This is not consistent with the rule of similarity of training and testing dataset network environments.

[18] is a recent work, which proposes multistage classifiers. Binary particle swarm optimization (BPSO) is a method applied by this work to properly select the best flow features. Three methods (port, payload, and statistical-based) are in-targeted into a multistage classifier. The working idea is very good; however, it was not tested as online classification which identifies traffic with capture speed. Another shortcoming is that the classifier can make its decision only based on the first stage (port based method).
III. ONLINE INTERNET TRAFFIC CLASSIFICATION ALGORITHM

A. General Concepts

Definition 1: Flow is a group of packets share the same 5-tuples (source address, destination address, source port, destination port, and transport protocol). Flow can be represented by TCP or UDP packets. We consider unidirectional flows, which defines client-server traffic as different from server client traffic. Definition 2: Real time traffic is the Internet traffic captured from the campus network during the period of experiments. Definition 3: Offline decision is the decision by the classifier about the flows identification, which is taken offline after capturing time. Definition 4: Online decision is the decision by the classifier about the flow identification, which is taken online within capturing time. Since the Internet applications are continuously being developed, it is difficult to classify the traffic by using only one classification method [22]. This paper develops online Signature Statistical Port Classifier (SSPC), which is making classification decision near to the capturing time. The classifier makes his final decision based on three parallel partial decisions (port classifier, signature classifier, and statistic classifier).

A.1 Port classifier

As mentioned in Section 1, port based classification cannot achieve a high accuracy all the time. In this paper port classification was used as a part of our classification system and it represents low priority of SSPC classification decisions. In most cases, SSPC classification decisions are not made based on the port classifier alone, but shared with the other two classifiers. We developed the port classifier algorithm as a part of the SSPC algorithm. The port classifier makes its own decision based on the port database. Easley, our port classifier algorithm, looks for the port number of the considered flow in the port database. If it is found, then the considered flow will be classified based on port classifier rules.

A.2 Signature classifier

Payload classification can achieve high accuracy, but it cannot work with encrypted traffic. As before, our classifier did not fully depend on payload, but it only represented a part of the SSPC decision. We developed a signature classifier algorithm, which is the second part of the SSPC algorithm; this algorithm makes its classification decision based on some saved signatures. We added very general signatures (such as DNS query and http host) for the considered applications, which are extracted from the application layer. If the considered flow carries a signature from the signature database, then it will classify based on the signature group.

A.3 Statistical classifier

The main problem that meets ML classification is the high false positive. To reduce this problem we consider two issues: firstly, the offline training datasets were continuously updated and collected manually from the same network we needed to classify, and secondly, the statistical classifier was supported by the other two classifiers. The statistical classifier algorithm is the third part of the SSPC algorithm. Same as before, ML classifiers represent a part of our system decision.

SSPC

For the purpose of increasing the classification efficiency, SSPC was proposed. SSPC is the result of the three previous classifiers’ decisions. Differing from the previous studies [14] and [18], SSPC did not base on hardware component; it also did not make its decision based only on one method. The online flow classification occurred after comparison of three stage classification decisions. Moreover, SSPC was tested for online classification decisions.

B. SSPC Architecture

Figure 1 illustrates the classifier stages, which start by fully packets capturing using traffic mirror.

![SSPC Architecture Diagram](image)

Before delivered to the three classifiers, the traffic was divided into flows based on the 5-tuples. Each flow will classify three times by each of the three classifiers. The port classifier (algorithm) will compare the captured flow port with a list of saved port numbers. If the captured flow belongs to any group of saved port numbers, it will be identified as its group. The second classifier (statistical) will work in parallel with the first classifier. Based on offline training and testing datasets, some classification rules were built. Based on these rules, the statistical classifier (algorithm) makes its online decisions to identify the captured flows. On the other hand, the signature classifier will classify the same flow at the same time as the previous two classifiers;
the classifier will compare a part of the captured flow with the signature database. If the signature matches any of the saved signatures, the classifier will make its online decisions to identify the captured flows. SSPC is an algorithm which compares between the three classifier results and makes its online classification decision based on some priorities rules.

C. SSPC Algorithm

The SSPC algorithm is shown in algorithm 1. Matlab version 7.5.0.342 (R2007b) was used to develop the algorithm. The SSPC algorithm consists of three partial classifier algorithms (described in section 4.2); each classifier has its own classification decision. Because of the accuracy of the signature classifier, the first priority in SSPC decision making goes to the signature classifier. If the signature classifier makes any decision about this flow, then SSPC final decision will to the decision made by the signature classifier. The second priority of SSPC happen in the case when all the partial classifiers cannot make any decision about this flow. In this case, SSPC will classify the flow as unknown. The third priority of SSPC occurs in the case when the statistics and the port classifiers have the same decision and signature classifier have no decision (unknown). In this case, SSPC identifies this flow based on the statistic and port classifiers. When the statistic and port classifiers have different opinions about the flow, SSPC will classify this flow as the statistic classifier as. The SSPC decision is based on the port classifier in only one case. This occurs when the port classifier has a decision whilst both statistic and signature classifiers have no decision about the flow. The pseudocode for SSPC algorithm is as follows:

```
% Define variables
Array port_DataBase;
Array signatures_DataBase;
string statistical_rules;
start packets capturing;
// divide captured packets into flows
if the packet belongs to an existing flow
then adds this packet to the existing flow
else
initializes a new flow;
end if
// run the three algorithm classifiers
for each traffic flow
{
% check signature
inspects n packets in the flow;
if class_signatures found
decision1=classify this flow according to signatures_DataBase;
else
decision1=classify this flow as unknown;
end if
% check statistic
if statistical_of_the_flow achieved any statistic_rules
```

IV. EXPERIMENTS AND ANALYSIS

In order to evaluate our methodology, several experiments were conducted. Real-time Internet traffic was collected from the campus network. This include four types Internet application class (WWW, FTP, Skype, online game). The WWW class includes http and https applications which have the higher percentage of the campus traffic. FTP traffic was generated by students to access ftp servers for resources download, which includes both data and control traffic. Skype has gained significant attention and has become one of the most popular forms of VoIP software, and it is used in our campus network. Skype traffic was generated by real communication sessions (calls) between Skype clients (SC), which are located within and outside the campus area. League of Legends (LOL) is the most widely played online game in our campus network, which belongs to Garena Plus online games. Table 1 show the flows considered for the four classes, which are obtained manually through the monitored clients (IPs). By using monitored IPs we ensure the training and testing datasets were collected from the same network without the need for standard (labeled) datasets.

<table>
<thead>
<tr>
<th>Class</th>
<th>Applications</th>
<th>Number of flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW</td>
<td>http, https</td>
<td>1857</td>
</tr>
<tr>
<td>FTP</td>
<td>FTP-data, FTP-control</td>
<td>304</td>
</tr>
<tr>
<td>Skype</td>
<td>Skype</td>
<td>2044</td>
</tr>
<tr>
<td>Online games</td>
<td>LOL</td>
<td>52</td>
</tr>
</tbody>
</table>
For ML training purposes, we captured traffic from some monitored clients. Offline ML classification was performed to select the optimum features and algorithm. After some filtering, rule.PART algorithm within Weka [19] was selected as the ML classifier; rule.PART rules were built into the statistical classifier algorithm. Inter-arrival time and packet length (size) are used as traffic features. From these two features, some statistics factors were calculated which are shown in Table 2.

<table>
<thead>
<tr>
<th>TABLE 2. SELECTED FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max of inter-arrival time</td>
</tr>
<tr>
<td>Min of inter-arrival time</td>
</tr>
<tr>
<td>Mean of inter-arrival time</td>
</tr>
<tr>
<td>Variance of inter-arrival time</td>
</tr>
<tr>
<td>Standard deviation of inter-arrival time</td>
</tr>
<tr>
<td>Max of packet length</td>
</tr>
<tr>
<td>Min of packet length</td>
</tr>
<tr>
<td>Mean of packet length</td>
</tr>
<tr>
<td>Variance of packet length</td>
</tr>
<tr>
<td>Standard deviation of packet length</td>
</tr>
</tbody>
</table>

Before going into online decision experiments, offline works were performed to validate the methodology. First, each classifier (port, signature and statistical) was used over each class dataset (table 1) and the result of each case was recorded. Second, in the same manner, SSPC algorithm was used over each class dataset separately. Table 3 and Figure 2 show the individual classifiers accuracy and SSPC accuracy. For each of five classes of the considered datasets, SSPC shows higher accuracy compared to the other partial classifiers.

<table>
<thead>
<tr>
<th>TABLE 3. OFFLINE CLASSIFICATION ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>WWW</td>
</tr>
<tr>
<td>FTP</td>
</tr>
<tr>
<td>Skype</td>
</tr>
<tr>
<td>LOL</td>
</tr>
</tbody>
</table>

In the online decision, the same offline applications through two different experiments were considered. Similar as in the offline experiments, the applications were run through the monitored clients. The testing dataset generated were totally different from the training dataset. As an example in some clients, we ran only WWW applications and then checked in parallel (at the same time) what the decision of each classifier was and what is SSPC decision.

<table>
<thead>
<tr>
<th>TABLE 4. NUMBER OF FLOWS FOR ONLINE DECISIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
</tr>
<tr>
<td>www</td>
</tr>
<tr>
<td>Skype</td>
</tr>
<tr>
<td>LOL</td>
</tr>
<tr>
<td>FTP</td>
</tr>
</tbody>
</table>

Table 4 shows the number of flows generated by each of the online experiments. Table 5 and Figure 3 illustrate the results of online decisions. The WWW classification accuracies are 90.15% and 88.34%, which is higher when compared with the other three classifiers. Skype signatures and port numbers were found only when considering the traffic of user’s login (experiment 1). This results show that SSPC is the higher accuracy between the other classifiers. However, when we did not consider login traffic (experiment 2), no signatures or well port numbers was found, which makes SSPC accuracy equal to statistics classifier. In the case of LOL online game, SSPC accuracy is higher than others, but it still low (68.78 & 87.59%). This is because of the low amount of datasets considered in the offline training stage. In FTP data, SSPC accuracy is acceptable but it less than port classifier. This because of two reasons, first: there is no FTP signature was added to support SSPC decision. Second: the considered FTP traffic was generated by some monitored clients, which used real FTP port numbers (20 & 21). The port accuracy expected to be low in case we have some clients used static IPs or VPN network. The last column in Table 5 shows the average of the classification time (in seconds) for each flow. As an example, classification of single WWW flow in Experiment 1 by SSPC was taken 0.01 seconds after the end of flow capturing.

<table>
<thead>
<tr>
<th>TABLE 5. ONLINE CLASSIFICATION ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
</tr>
<tr>
<td>www</td>
</tr>
<tr>
<td>Skype</td>
</tr>
<tr>
<td>LOL</td>
</tr>
<tr>
<td>FTP</td>
</tr>
</tbody>
</table>

Figure 2 Offline classifiers accuracy
V. Conclusion

Port based classification has the advantage of non-complexity; however, it cannot achieve high accuracy with applications of unknown port numbers. On the other hand, payload classifiers have the advantage of accurate classification, but incapable to classify encrypted traffic. Statistical classifier has the benefit of classifying encrypted traffic, but it has the problem of high false positive and features overlapping. In this paper, signature statistical port classifier (SSPC) algorithm for online traffic classification was proposed. In parallel, each of the three partial classifiers (based on the three methods) makes a decision about each traffic flow. SSPC based on some priority rules calculates the final decision from the outcomes of the three classifiers. Real time datasets (more than 7900 flows) were captured in the campus environment, which includes WWW (http, https) and non-WWW (FTP data, FTP control, online gaming, and Skype) traffic. The SSPC system was tested in two stages, offline and online. The results of the offline experiments show that SSPC has higher accuracy compared to the three partial classifiers. To further validate the robustness of the algorithm, online classification using SSPC algorithm was carried out. The classification decision was made immediately after the end of flow capture. Thus, SSPC has achieved the two objectives; the first, capitalizing on the effectiveness of the individual classifiers which sums up to effectively increase SSPC accuracy. Second, SSPC can still classify the online Internet traffic without any compromise in delay.

REFERENCES


