

Using host profiling to refine statistical application identification

Mohamad Jaber, Roberto G. Cascella and Chadi Barakat
INRIA - France

Email: {mohamad.jaber, roberto.cascella, chadi.barakat}@inria.fr

Abstract—The identification of Internet traffic applications is very important for ISPs and network administrators to protect their resources from unwanted traffic and prioritize some major applications. Statistical methods are preferred to port-based ones and deep packet inspection since they don't rely on the port number, which can change dynamically, and they also work for encrypted traffic. These methods combine the statistical analysis of the application packet flow parameters, such as packet size and inter-packet time, with machine learning techniques. Other successful approaches rely on the way the hosts communicate and their traffic patterns to identify applications.

In this paper, we propose a new online method for traffic classification that combines the statistical and host-based approaches in order to construct a robust and precise method for early Internet traffic identification. We use the packet size as the main feature for the classification and we benefit from the traffic profile of the host (i.e. which application and how much) to refine the classification and decide in favor of this or that application. The host profile is then updated online based on the result of the classification of previous flows originated by or addressed to the same host. We evaluate our method on real traces using several applications. The results show that leveraging the traffic pattern of the host ameliorates the performance of statistical methods. They also prove the capacity of our solution to derive profiles for the traffic of Internet hosts and to identify the services they provide.

I. INTRODUCTION

The identification of Internet traffic applications is very important for ISPs and network administrators to protect their resources from unwanted traffic and prioritize some major applications. On the one hand, this allows to treat flows in a different way based on their quality of service requirements and allocate more resources based on the type of traffic. On the other hand, it can serve for security reasons by blocking unwanted traffic and limiting worm spreading or looking closely at those users who run non legacy applications.

The identification of Internet traffic becomes more and more complex because of mechanisms that bypass firewalls or mask the type of application. Historically, the recognition was done by using the port number. Yet, some applications use dynamic non-standard port numbers; this is typically the case of telephony over IP. Other applications hide themselves using standard ports stolen from other applications, such as port 80, to pass firewalls. These ports are usually given by the end host and thus they can be easily changed.

Current techniques of "Deep Packet Inspection" (DPI) [1], [2] make it possible to go further in the identification of the applications but they require a complete and costly exploration

of the payload of the packets. This induces an important load to inspect packets and create the signatures, which requires updates with the appearance of new applications. Moreover, when packets are encrypted, the recognition fails.

The statistical techniques [3]–[5] seem to be today an interesting alternative. They allow to recognize and to classify the applications according to their statistical signatures. These signatures can be volumes (number of bytes) per connection, connection durations, rates, inter-packet delays, packet sizes, and direction. Most of these techniques require a machine learning phase to perform the classification of connections (or flows) into applications. In [3], Bernaille et al. test three clustering algorithms (K-Means, Gaussian mixture model, and the Spectral clustering); the input features to assign flows to applications are the size and the direction of the first four packets jointly used. In [4], Crotti et al. classify Internet traffic by using the packet size and inter-packet time. In our previous work [5], we develop a method to iteratively classify Internet traffic while using the size and the direction of the packets.

The common feature of statistical methods is that they classify every flow independently of each other using the pattern of its packets (size, time, and direction). Indeed, they don't correlate the information across flows having as end points the same hosts, thus not using any information about the traffic pattern of the originating host or the type of services that run on the destined server. Some recent works have focused on this aspects by considering the role of the hosts [6] or the relations of the traffic between end points [7]. Our solution differs from these studies since we only rely on the information that a monitor collects passively from the packet flows and we do not require any information related to the groups of communicating hosts, such as a *graphlet* [6].

We believe that the classification of previous flows sharing the same IP address either as source and/or destination is important to refine the classification of future flows. For instance, a host browsing the Web is more prone to open several consecutive HTTP connections. A machine hosting a POP3 mail server is very likely to receive POP3 flows. In general, hosts have profiles for their flows either because of the behavior of users or the services run on them, and these profiles can help in the identification of flows in which they are implied. Our idea is to build the traffic profile of hosts, based on the result of the classification of previous flows, and then use this information to refine the classification of subsequent flows. On one hand these profiles help in flow classification

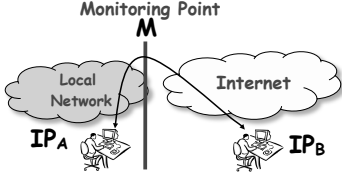


Fig. 1: The system.

and on the other hand they point to the behavior of the users behind them and on the network services they deploy.

In this paper we propose a novel two-step approach to affect flows to applications. In the initial classification phase, we use an iterative statistical technique to classify Internet traffic, based solely on the flow statistical features. The results give an initial classification. In the second phase, we use the traffic profile of the host to refine the classification and, then, to update the host profile based on the classification results. Our contribution can be summarized as follows. First, we define the host profile and we determine the host-based probability that a flow is of a given application. We then develop a new method that relies on the result of the classification of flows from the same host to determine the profiles of hosts; these profiles are later used as an initial guess before the classification of future flows. The host profiles are updated after each classification using an exponential weighted moving average filter to absorb any transient behavior; the way the profile accounts for past classified flows depends on a discounting parameter, which can be decided by the network administrator.

The rest of the paper is organized as follows. Section II introduces the host profiling. Section III explains our classification method. Section IV and Section V describe the traces and the evaluation results, respectively. Section VI concludes the paper.

II. HOST TRAFFIC PROFILE

In this section we introduce our definition of the traffic profile of a host and we present the benefits of using this information to refine the classification of Internet applications while discussing the related work. The methodology herein described is general and can be integrated to any flow classification method transparently.

Without loss of generality we consider a monitoring point at the edge of the network, located in the ISP network, as shown in Fig. 1. The monitor *passively* captures the flows between any two hosts; a flow consists of the packets with the same 5-tuple (IP source and destination, port source and destination, IP protocol). For each flow there are one host located inside the ISP network (IP_A in Fig. 1) and a destination host downstream the monitoring point (IP_B in Fig. 1); we don't assign any specific role to these two hosts, IP_A and IP_B , which can act as client or server during a session indifferently. The monitor inspects the packets of each flow and extracts statistical information, or the signature of the flow, such as packet size, inter-packet time, direction of the packet, etc. This statistical signature is then used to assign the flow to the application that matches it. In this section we focus on the

profile of the host, while the definition of this signature and classification procedure are detailed in Section III.

The traffic profile of a host consists of the type of applications which run at the host and generate Internet traffic. This profile is determined at the monitor, which stores the results of the classification of the Internet traffic of the hosts. Practically, the monitor can be interested to log the traffic of the hosts inside the ISP network, and/or only those of interest from outside the ISP network. In addition, the monitor might decide to store information about some IP addresses that run dedicated services since this can help with the classification of Internet flows. The traffic profile, so computed, gives an indication of the preferred applications that run at the host and of the type of traffic the ISP would expect from the host.

The motivation behind our solution is the recent studies on residential networks, which give an insight of user traffic profile [8], [9]. An interesting outcome of these studies is that users tend to hardly mix P2P and HTTP (Web streaming), which are the most predominant applications [8].

In this section, we first discuss how a monitor computes the probability that a flow of packets between two hosts is of a certain application solely using the traffic patterns of these hosts. Then, we discuss how the monitor computes and updates the host profile.

A. Host based probability of a flow

The host based probability of a flow is defined as the probability that a flow is generated by an application computed based on the traffic profile of the hosts, i.e., source and destination. If we consider that the two traffic profiles of the source and destination of a flow are different and that these are used jointly in the computation, then, this probability consists of those cases when the predictions computed with the partial info of each host are in accordance.

Let F denote a function that associates a packet flow between a source S and destination D to an application $A(i)$, with $1 \leq i \leq N_A$ and N_A being the number of monitored applications. Thus, F_S and F_D are the functions that assign the flow to the application A_S and A_D based solely on the traffic profile of the source and destination respectively. Then, let $P(F_S = A_S|S)$ (or $P(F_D = A_D|D)$) be the probability that, given the host traffic profile of the source, the flow is of an application A_S (or A_D for the destination). The probability $P(F = A(i))$ that the flow is of application $A(i)$ can then be computed as follows:

$$\begin{aligned} P(F = A(i)) &= P((F_S = A(i)_S) \cap (F_D = A(i)_D) | A_S = A_D) \\ &= \frac{P(F_S = A(i)|S) * P(F_D = A(i)|D)}{\sum_{j=1}^{N_A} P(F = A(j)|S \cap D)} \quad (1) \\ &= \frac{P(F_S = A(i)|S) * P(F_D = A(i)|D)}{\sum_{j=1}^{N_A} P(F_S = A(j)|S) * P(F_D = A(j)|D)} \end{aligned}$$

Eq. (1) shows that we compute the probability by considering the cases when the prediction for each host is in accordance by considering the traffic profiles of S and D separately, i.e., we know that the same application is running

on both sides. Equation (1) also holds when the monitor only records the traffic profile of one of the two hosts. In fact, if we assume a uniform probability for the other host, e.g., $P(F_D = A_D|D) = \frac{1}{N_A}$, then, equation (1) simplifies to $P(F = A(i)) = P(F_S = A(i)|S)$.

B. Host profile definition and update

The monitor computes and updates the profile of the hosts. After capturing and classifying the flows, two traffic profiles are generated for each host. Indeed, each host can be the source or the destination of the Internet flows. The former is the host that sends the first packet of the flow, as we discuss in Section II, while the latter is the one that receives it. We keep these two profiles separated since they characterize the role of the host when being the source or the destination. For example, a host can send HTTP requests to a server or receive SSH requests when is running a local SSH server. In the rest of the section, we consider a generic host and we focus on the computation of the source profile for this host; the destination profile is defined in the same way.

Let S denote the generic source host of a flow and F_S the function that maps the flow to an application by only leveraging the traffic profile of the source. The monitor computes the host profile by using previous classified flows. The profile, denoted $P(A|S)$ in this case, is defined as the prior distribution for the flows in the space \mathcal{A} , which defines the applications $A(i)$, $1 \leq i \leq N_A$. If the monitor has not any information about previous traffic of a host, then, the monitor considers a uniform prior distribution. The prior distribution is updated after each classification of a new collected flow.

The profile update works as follow. Let $P_{(n-1)}(A(i)|S)$ be the prior probability for application $A(i)$ computed from the past $(n-1)$ flows. The monitor affects the n -th flow to the application $A(i)$ with probability $P(F_S = A(i)|S)$ for each application. Then, the posterior probability for each application is computed as follows:

$$P_{(n)}(A(i)|S) = \lambda * P_{(n-1)}(A(i)|S) + (1 - \lambda) * P(F_S(n) = A(i)|S). \quad (2)$$

$P(F_S(n) = A(i)|S)$ is the result of the classification of flow n and λ , $0 \leq \lambda \leq 1$, represents the discounting factor for past classifications. When λ is close to 0, the profile is computed by associating a higher weight to the most recent flows. When λ is close to 1 the profile is calculated over a longer period, which means that the profile is determined in equal measure by all previous classified flows. When $\lambda = 1$ the profile corresponds to the initial prior distribution, which in our case assigns a uniform probability to all applications. The best choice of λ depends on the traffic pattern of the host and on the performance of the classifier. We will discuss more about λ in Section V.

The traffic profile of the same host while being the destination, it is computed in a similar way by considering only the flows destined to this host. It is worth noticing that the monitor needs to store the two prior distributions if it want to fully determine the profile of the host. In practice, given the

limitation of the resources, the monitor can decide to track and store profiles for a subset of hosts (source and/or destinations) and use simple uniform profiles for the other hosts. In this case, the method will also work well but with less accuracy since the more hosts we track better the classification of Internet flows is for these hosts. Table I shows an example of the source and destination profiles of a host.

TABLE I: Example of a traffic profile of a host

Applications:	FTP	HTTP	POP3	SMTP	SSH
Source:	0.02	0.76	0	0.2	0.02
Destination:	0.22	0	0.1	0.23	0.45

III. METHOD DESCRIPTION

Our purpose for the classification of Internet traffic is to detect online which flow belongs to which application. We use a statistical and iterative method that computes the probability that packets are generated by an application. We have defined and used this method to classify Internet traffic based on the size of the packets in [5]. The method allows an iterative classification of the flows for each packet size independently and uses more packet sizes for the identification of an application until the classifier reaches a predefined threshold. Each flow corresponds to a sequence of N packets independently of their direction.

In this section we first propose an overview of our method and then we detail how the method uses the host profile to refine the classification. The method consists of three main phases: the model building phase, the classification phase, and the application probability or labeling phase. The the traffic profile of the host is used in the labeling phase.

A. Model building and classification phases

We use K-Means as supervised machine learning algorithm to partition the input in a predefined number of clusters. Given the number of clusters N_C , K-Means assigns each input feature to a cluster so as to minimize the Euclidean distance of each input from the centroid of the cluster.

Pkt_k denotes the packet size, i.e., the observations, and for each packet size we train separately K-Means to obtain different set of classes. Thus, the packet sizes of position k have their own independent training, and the model used for testing its determined by the position of a packet in a flow. The input feature corresponds to the size of the packet associated with a sign that represents the direction of the packet. A positive sign corresponds to a packet from the source to the destination. In the learning algorithm, every class is affected by all applications with different probabilities proportional to the number of flows from each application present in the class. Hence, each class defines the probability that the elements within this class are generated by the applications.

The model building phase consists of constructing these sets of classes (clusters) by using a training data set, described in Section IV. Let denote $C(j)$ the clusters, where $1 \leq j \leq N_C$ and N_C is the number of clusters. Then, the per-class probability $P(C(j)|A(i))$, knowing the application $A(i)$ is computed

for all the clusters during this learning phase. We build a separate model, i.e., set of classes, for every packet size noted by Pkt_k and we use these classes for the classification phase.

The classification phase consists of using the classes defined in the learning phase to test and assign the Internet flows to a class. The test is performed by computing the Euclidean distance between the input feature from the k -th packet in the flow and the centroid of each class determined for the k -th packet size. We affect the point to the closest class. The test is repeated for all the packet sizes of a flow iteratively until we reach a predefined threshold. The classification result is the probability that the packet size Pkt_k identifies an application.

B. Application probability or labeling phase

In the labeling phase we assign a flow to an application knowing the result of the classification and the host based probability computed from the profiles of the source and destination, as discussed in Section II. We combine iteratively the results of the classification for each single packet size and we calculate the probability ($P(A(i))$) that a flow belongs to an application $A(i)$ given the prediction from the host profiles and the classification results of the first N packet sizes (i.e., class $C(j(1))$ for the first packet size, class $C(j(2))$ for the second packet size and so on).

$$\begin{aligned}
 P(A(i)) &= P(A(i)|Result) \cap P(F = A(i)) \\
 &= \frac{P(F = A(i)) * \prod_{k=1}^N P(C(j(k))|A(i))}{\sum_{i=1}^{N_A} [P(F = A(i)) * \prod_{k=1}^N P(C(j(k))|A(i))]} \quad (3)
 \end{aligned}$$

$P(F = A(i))$ is the probability that a flow between a source and a destination comes from application $A(i)$ based on their traffic profiles and it is calculated in Eq. (1). $P(C(j(k))|A(i))$ is the probability that Pkt_k of a flow belongs to the class $C(i)$ knowing the application $A(i)$. N_A is the total number of applications. We call $P(A(i))$ the *assignment probability*. It combines the result of the classification, obtained with the K-Means clustering method, and the result of the classification that one would have if solely the pattern of the hosts is used to predict the type of application for the next flow.

This assignment probability is computed when the monitor captures each packet of the same flow. This means that the classification of the application starts with the first packet. This iterative process stops when the highest assignment probability is above a predetermined threshold or the maximum allowed number of tests is reached. The threshold is seen as a way to leave the classification phase earlier when one is sure about the type of application. The monitor updates the profiles of the hosts that are of interest, i.e., the source and/or the destination, once the labeling phase ends. The host profiles are updated as described in Section II-B.

IV. TRACE DESCRIPTION

In our analysis we use two real traces, see Table II for details. The two traces have been collected at the edge gateway of the Brescia University's campus network. The first trace, noted

TABLE II: Traces Description

Source and Date	Application	training	testing
Brescia University April 2006 [4]	HTTP	8000	17,263
	SMTP	8000	19,835
	POP3	8000	19,935
Brescia University Fall 2009 [10]	HTTP	500	30422
	HTTPS	500	3608
	EDONKEY	500	3702
	BITTORENT	500	3608

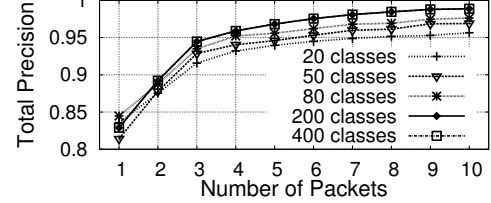


Fig. 2: Total precision of an Internet traffic classification with a different number of class as input to K-Means (Trace I).

trace I [4], was collected during April 2006 and the second trace, noted trace II [10], was collected on three consecutive working days during fall 2009. Every trace consists of two sets, a training set and a testing set; the type of applications associated with each flow is determined with a deep packet inspection method. In the learning phase we use the training set, which consists of an equal number of flows per application to ensure that there is no bias in our learning. The application flows in the training set are only used to construct the classes in K-Means. The testing set is used to evaluate how well our iterative method behaves in identifying the application.

V. EXPERIMENTAL RESULTS

In this section we present the evaluation results of our method when the traffic profile of the hosts is used to refine the classification. We use the traces described in Section IV and we profile the hosts with the same IP prefix, i.e., those inside the Brescia campus. For addresses outside the campus, we have counted an average of 10 flows per IP address, therefore there is not a significant number of flows per IP to compute the profile. The flows are all TCP connections and the hosts within the campus are the source of the flow. We evaluate our method by showing the overall Precision over all applications. We define the precision as the ratio of flows that are correctly assigned to an application, $TP/(TP + FP)$. The overall precision is the weighted average over all applications given the number of flows per application. Where the True Positive (TP) rate is the percentage of flows of application I correctly classified and the False Positive (FP) rate is the percentage of flows of other applications classified as belonging to an application I .

We run the test for all the available packet sizes to test its significance as a feature for identifying applications. We set the number of clusters equal to 200 for K-Means which has shown the best results (see Figure 2).

A. Classification results

In this section we discuss the performance of the classification method when the host profile is used to refine the

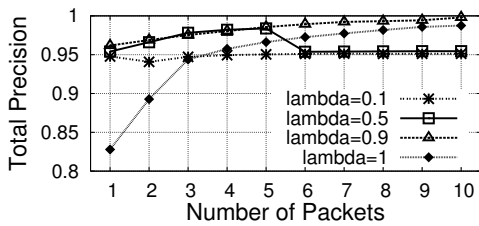


Fig. 3: Total precision versus the number of packets (Trace I).

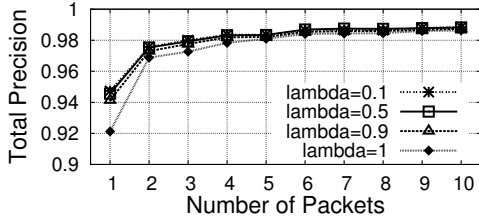


Fig. 4: Total precision versus the number of packets (Trace II)

probability that a flow is of a given application type.

Fig. 3 and 4 plot the total precision for trace I and trace II respectively versus the number of packets used for the classification. Our method classifies a flow at each packet iteratively, as we discuss in Section III. The different lines in the plot correspond to the precision of the classifier when different values of the discounting factor λ are used. The value of λ determines the weight assigned to the last classification results. When $\lambda = 0.1$, the most recent classification results characterize the profile of the host. When $\lambda = 0.9$, the host profile is computed over a longer period. The value of $\lambda = 1$ means that a uniform probability is associated to each application, thus, the host profile is not used, as we discuss in Section II-B. The results show that the precision of the classifier improves considerably when the profile of the hosts is used to decide in favor of this or that application, especially for the first four packets.

For Trace I, we can observe in Fig. 3 that a value of $\lambda = 0.9$ gives the best performance for the classifier. We obtain a precision of 96% already after two packets reaching 99.9% after 10 packets. For $\lambda = 0.1$ and 0.5 the classifier predicts with less accuracy the applications with a precision that reaches 95%. With this value of λ the classifier is more sensitive to recent flows. Thus, the classification is less accurate if the host has a uniform traffic behavior over all applications. For this trace we have that a big number of flows belong to two different applications and are generated uniformly by the same host, and that the method classifies the applications with less precision for small values of λ . If one does not leverage the host profile ($\lambda = 1$) the precision of the classifier is quite low (89%) after two packets, but it keep increasing when more packets are analyzed (98% after 10 packets). This result confirms that the host profile helps in deciding about a flow when little information can be extracted from the statistical analysis of the flow.

For Trace II and for all the selections of λ , we have better performance compared to the classification without host profile information ($\lambda = 1$). Indeed, Fig. 4 shows that if the

host profile is used, then the precision already increases after the first packet, and then converges to 99% after the fifth packet in all cases. These results show that the use of the host information increases the precision (in comparison of the classification without host information) of the classifier especially for the first four packets. We can conclude that the profile of the host gives an early characterization of a flow because of the traffic pattern of the host. For instance, we can consider that a host browsing the Web has high probability to have a sequence of HTTP connections. Thus, the use of information about the host profile helps our statistical method.

VI. CONCLUSION

In this paper we present our new method for Internet traffic identification that combines the statistical and host-based approaches. The statistical parameters that we use are the size and direction of the first N packets. The novelty of our approach consists in leveraging the host profile to refine the classification. First we define the profile of the host and how it is updated. Then we show how the profiles of the source and destination hosts are used to assign a prediction probability to the new flow. We evaluate our solution on two real traces. We profile the hosts and we test our method for different values of the discounting factor λ , which defines how the profile accounts for past flows. The results show a great improvement for the classification of applications when the host profile is used. In particular, the classifier reaches a precision of 98% after using 10 packets for the classification. For more details about the results of our method see [?].

REFERENCES

- [1] A. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *PAM*, October 2005.
- [2] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," in *WWW 2004 Conference*, Philadelphia, USA, May 2004.
- [3] L. Bernaille, R. Teixeira, and K. Salamati, "Early application identification," in *ADETTI/ISCTE CoNEXT Conference*, December 2006.
- [4] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," in *ACM-Sigcomm Computer Communication Review*, vol. 37, January 2007, pp. 5–16.
- [5] M. Jaber and C. Barakat, "Enhancing application identification by means of sequential testing," in *NETWORKING*, Aachen, Germany, 2009.
- [6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: Multilevel traffic classification in the dark," in *SIGCOMM*, New York, USA, 2005.
- [7] Y. Jin, N. Duffield, P. Haffner, S. Sen, and Z.-L. Zhang, "Inferring applications at the network layer using collective traffic statistics," in *the 22nd International Teletraffic Congress (ITC 22)*, 2010.
- [8] M. Pietrzyk, L. Plissonneau, G. Urvoy-Keller, and T. En-Najjary, "On profiling residential customers," in *TMA*, Vienna, Austria, 2011.
- [9] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband internet traffic," in *IMC*, 2009.
- [10] T. II, "Brescia university," <http://www.ing.unibs.it/ntw/tools/traces/>.