

# Boosting statistical application identification by host profiling

Mohamad Jaber, Roberto G. Cascella and Chadi Barakat

EPI Planète, INRIA, France  
Email: {mjaber, rcascell, cbarakat}@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 and deep packet inspection techniques since they don't rely on the port number and they work also for encrypted traffic. These methods combine the statistical analysis of the flow parameters, such as packet size and inter-packet time, with machine learning techniques. Other approaches rely on the interactions of the hosts and traffic patterns of host behavior to identify applications.

In this paper we propose a new online method for traffic classification that combines the statistical and host-based approach 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 to decide in favor of this or that application. This profile is updated online based on the result of the classification of previous flows originated or addressed to the same host. We evaluate our method on real traces. The results show that by leveraging the traffic pattern of the host ameliorates the performance of statistical methods.

**Keywords:** Internet traffic identification, statistical methods, Host profiling

## 1 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. On the other hand, it can serve for security reasons by blocking or looking closely at those users who run non legacy applications.

The identification of Internet traffic becomes more and more complex. Historically the recognition was done by using the port number [1]. But some applications now use dynamic port numbers; this is typically the case of telephony over IP. Other applications hide themselves using standard ports stolen from

other applications. These ports are usually given by the end host and thus they can be easily changed.

Current techniques of "Deep Packet Inspection" (DPI) [2, 3] 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 and requires updates with the appearance of new applications. Furthermore, when packets are encrypted, the recognition is not possible.

The Statistical techniques [4–8] 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 the approaches that use statistical features to identify applications require a machine learning technique to do the classification. In [5], McGregor et al. show the utility of using clustering algorithms for the identification of the traffic. They propose to use an unsupervised machine learning, called auto class, and the following statistical criteria: packet size, inter-arrival time, byte count, and connection duration. In [4], Moore et al. use a Naive Bayesian classifier for TCP traffic, and they try to find the best set of statistical criteria. In [6], Bernaille et al. test three clustering algorithms (K-Means, Gaussian mixture model, and the Spectral clustering) and the input features to assign flows to applications are the size and the direction of the first four packets jointly used. In [7], Crotti et al. classify traffic while using the packet size and inter-arrival time. In our previous work [8] we developed a method to iteratively classify traffic while using the size and the direction of the packets.

However, the major problem of these statistical methods is that they classify every flow independently. Indeed, they don't use any information about the traffic patten of a host or the type of services that run on a server. We believe that the classification of previous flows is important to refine the classification of Internet applications. A host browsing the web is more prone to open several consecutive HTTP connections. In this paper we propose to profile a host based on the result of the classification of previous flows and then use this information to refine the classification.

Previous approaches [9, 10] that considered the role of the host differ from ours because we focus only on the information that a monitor can collect passively from the flows. In [10] Trestian et al. characterize the end-point behavior by collecting publicly available information on the web based on the IP address of the host. BLINC [9] is a solution for Internet traffic identification that considers the role of the hosts. It focuses on the source and destination hosts of the flows to determine the host behavior which is studied across three levels: social to account for the host popularity and communities of hosts (groups of communicating hosts), functional to identify the functional role of a host (offer services, use services), and protocol patterns of the hosts. The main difference, which is also the strength, of our approach consists of only considering the interactions between hosts that are detected by the monitor. We construct and leverage the

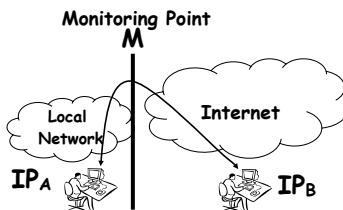


Fig. 1: The system.

profiles of the communicating hosts simultaneously but we do not require the traffic monitor to maintain a detailed history of their interaction.

Our contribution in this work can be summarized as follows. First, we define the host profile and we determine host based probability associated to a flow that is of a given application. We develop a new method that relies on the result of the classification of Internet applications to determine the profile of the host. The main idea is to combine the statistical properties of a flow with the traffic profile of the end-points to associate flows to applications. These profiles are updated after each classification with an exponential moving average; the way the profile accounts for past classified flows depends the discounting parameter, which can be decided by the network administrator. Second, we use a real trace to test our method and we show how to characterize the traffic pattern of a host.

The rest of the paper is organized as follows. Section 2 discusses the profile of the host. Section 3 explains our classification method. Section 4 and Section 5 describe the traces and the evaluation results respectively. Section 6 concludes the paper.

## 2 Host traffic profile

In this section we discuss how we determine the traffic profile of a host and what are the benefits of using this information to refine the classification of Internet applications. The methodology herein described is general and it can be integrated to any classification method transparently. In our model and 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 users; a flow consists of the packets with the same 5-tuple (IP source and destination, port source and destination, IP protocol). For each flow, we consider the two end points: a 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 the hosts,  $IP_A$  and  $IP_B$ , which can act indifferently as client or server during a session. The monitor inspects the packets of each flow and extracts statistical information, such as packet size, inter-packet time, direction of the packet, etc. This information is used to create the signature of the flow and to assign the flow to the application that matches the signature. We will discuss in Section 3 the definition of this signature and classification procedure.

The traffic profile of a host is defined based on type of previous flows. This requires that the monitor collects statistical information about a flow, classifies the flow, and stores the result of the classification to track the activity of a host. In a real setting, we assume that the monitor logs only the traffic for the hosts inside the ISP network, or those of interest, and might store information about some IP addresses that runs dedicated services. The traffic profile, so computed, gives an indication of the preferred applications that run at the host.

The novelty of our approach consists of using this traffic pattern to predict future flows that involve the same host. 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.

## 2.1 Host based probability of a flow

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  $N_A$  the number of monitored applications. Let  $P(F_S = A_S|S)$  (or  $P(F_D = A_D|D)$ ) be the probability that, given the host traffic profile, the flow is of an application  $A_S$  for the source (or  $A_D$  for the destination). Then, the probability  $P(F = A(i))$  that the flow is of application  $A(i)$  is computed as follows:

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

Equation (1) means 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. 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  $P(F = A(i)) = P(F_S = A(i)|S)$ .

## 2.2 Host profile definition and update

We now discuss how the monitor computes the host profile and updates this information. Each host can be source or destination whether it does send the first packet of the flow. The monitor captures the flows and decides about a flow by using the source or destination profile of the host. This results in two traffic profiles for the same host. In the rest of the section, we discuss a generic host and 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. The monitor computes the host profile by using previous classified flows, in this case when the host is the source of the first packet. The profile  $P(\mathcal{A}|S)$  is thus defined as the prior distribution for the flows in the domain  $\mathcal{A}$ , 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 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 that the monitor affects 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 for flow  $n$  and  $\lambda, 0 \leq \lambda \leq 1$ , represents the discounting factor for past classifications. When  $\lambda$  is close to 0, the profile is computed by associating a higher weight to the most recent flows. When  $\lambda$  is close to 1 the profile is calculated over 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  $\lambda$  depends on the traffic pattern of the host and on the performance of the classifier. We will discuss more about  $\lambda$  in Section 5.

Table 1: 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

The amount of information that the monitor maintains to update the profile of the host is limited to the two prior distribution. Table 1 shows an example of source and destination profiles of a host.

### 3 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 [8]. 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  $Pkt_k$ , where  $k$  indicates the position of the packet in the flow independently of its 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 which are detailed in the following sections: the model building phase, the classification phase, and the application probability or labeling phase.

### 3.1 Model building and classification phase

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 Euclidian 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. The input feature corresponds to the value of the size of the packet associated with a sign that represents the direction of the packet. A positive sign corresponds to a packet from to 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 consists of constructing these sets of classes (clusters) by using a training data set, described in Section 4. This learning phase is used to compute  $P(C(j)|A(i))$ , i.e., the per-class probability, knowing the application  $A(i)$ . The probability is computed for all clusters  $C(j)$ , where  $1 \leq j \leq N_C$  and  $N_C$  is the number of clusters. 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 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 Euclidian distance between the input feature from  $Pkt_k$  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 consists in the probability that the  $Pkt_k$  identifies an application and it is given as input to a labeling function described in the following section.

### 3.2 Application probability or labeling phase

In the labeling phase we assign a flow to an application knowing the result of the classification and the probability computed from the profiles of the source and destination, as discussed in Section 2. 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.,

Table 2: Traces Description

Source and Date	Application	# of training flows	# of testing flows
Brescia University [7] April 2006	HTTP	8000	17,263
	SMTP	8000	19,835
	POP3	8000	19,935

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.  $P(C(j(k))|A(i))$  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 this probability the assignment probability. It combines the result of the classification, obtained with the K-Means clustering method, and the pattern of the hosts which give an indication of the next type of application flow. The assignment probability is computed when the monitor capture every packet size. Thus, we do not require to wait for a given number of packets to start the classification procedure. We stop this iterative process when the highest assignment probability is above a predetermined threshold or the maximum allowed number of tests is reached. This way the threshold is seen as a way to leave the classification phase earlier when we are sure about the flow. The monitor updates the profiles of the hosts, both the source and the destination, when the classification ends and it has assigned a flow to a given application. The update of the profiles is done as described in Section 2.

## 4 Trace Description

In our analysis we use one real trace, see Table 2 for details. The trace has been collected at the edge gateway of the Brescia University’s campus network [7] and specifies the sizes of the packets after the TCP connection has been established. The trace consists of two sets; the training set was collected during the first week in April 2006 and the testing set after two weeks. The training set indicates the type of application associated with each flow, which the authors of [7] 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 phase. During this phase we do not compute the host profile and the application flows are used only to construct the classes in K-Means. The host profile is only computed during the testing phase. We initially consider a uniform prior

distribution for the source and destination profiles of an unknown host and then we update the profiles once flows of this host are collected. The testing set is used to evaluate how well our iterative method behaves in identifying the application.

In our previous work [8], we use another data set collected at INRIA to test the classification method based on the packet size. We do not present any result from this data set because the flows were collected at the interface of the servers reserved by INRIA to unique applications. The profile of these hosts clearly shows this traffic pattern which has no value to test the proposed our host based classification method.

## 5 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 consider the case of a monitor that maintains the profile of the hosts that are located inside the ISP domain, since it is interested to understand what is the usage of the network by the ISP customers. In a real setting, the monitor might also decide to maintain the information about popular Internet servers and use their profile to compute the probability that a flow is originated by an application, as we discuss in Section 2. We use the trace described in Section 4 and we profile the hosts with the same IP prefix, i.e., those inside the Brescia campus. We have counted an average of 10 flows per IP address outside the Brescia campus and we have decided to not show the results since there is not a sufficient number of flows per IP to compute the profile. Thus, we only compute the client profile for these machines. The flows are all TCP connection and the hosts within the campus initiate the connection.

The metrics used for the evaluation are:

- *False Positive (FP) rate* is the percentage of flows of other applications classified as belonging to an application  $I$ .
- *True Positive (TP) rate* is the percentage of flows of application  $I$  correctly classified.
- *Precision* is 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.

We run the test for all the available packet size to test its significance as a feature for identifying applications. We set the number of clusters equal to 400 for K-Means. We have tested the supervised machine learning algorithm with different number of clusters and 400 gives the best results as it allows to group the features in small clusters and account for possible noise in the observations.

### 5.1 Packet size distribution

We initially analyze the first 10 packet sizes of the testing flows to understand how well the packet size characterizes an application. Fig. 2 and Fig. 3 show the

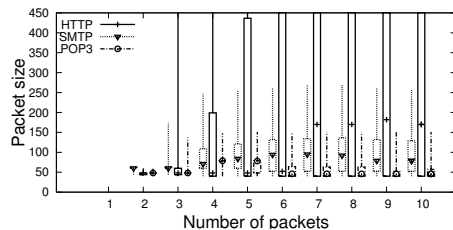
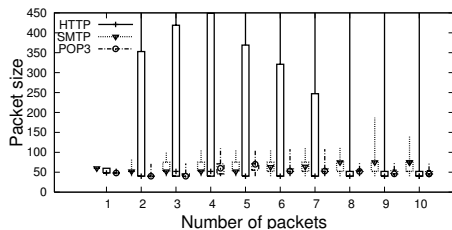


Fig. 2: Distribution of the size of packets sent by the hosts inside the campus      Fig. 3: Distribution of the size of packets received by the hosts inside the campus

distributions for each packet number, where the number indicates the order in a flow; we limit the plot to 450 bytes. The results from the packets sent and received by the hosts inside the Brescia campus are shown in Fig. 2 and Fig. 3 respectively. For each packet number, the plots show the median, and the bars indicate the quartiles, and the 2nd and 98th percentiles. From the distribution we first notice that all the testing flows are initiated by the hosts with the same IP prefix because there are no first packets of a flow in Fig. 3.

Fig. 2 shows that the first packet for the three applications has the same size in almost all the flows. In particular, the SMTP packet is larger than the others. POP3 and HTTP have the same value for the median but the latter spans more values for the size of the packet, shown by the 75th percentile. The second packet has a similar distribution for the three applications. In Fig. 3, the second packet has a size equal to the median and POP3 and HTTP have a similar distribution. By analyzing the first packets we can conclude that it is possible to distinguish an SMTP application from a POP3 or HTTP flow. From the 6th packet, the three applications have different distributions for the packet size. This means that it is possible to differentiate one application from the others both for packets sent or received by the hosts. We can conclude from these two figures that we using the packet size can be a very good parameter to differentiate between Internet applications.

## 5.2 Classification results

In this section we discuss the performance of the classification method when the host profile is used to refine the probability that a flow is of a given application type.

**Precision** Fig. 4 plots the total precision of the three applications versus the number of packets used for the classification. Our method classifies a flow at each packet iteratively, as we discuss in Section 3. The different lines in the plot correspond to the precision of the classifier when different values of the discounting factor  $\lambda$  are used. The value of  $\lambda$  determines the weight assigned to the last classification results. When  $\lambda = 0.1$ , the most recent classification results determines 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 have discussed in Section 2.2.

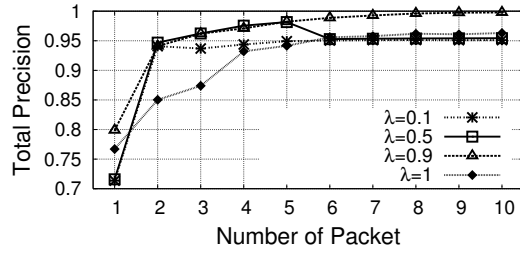


Fig. 4: Total precision versus the number of packets.

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. We notice that the precision increases when the classifier uses more packets to identify the flow up to packet 6 with  $\lambda = 0.5$ . This decrease depends on the type of flows that have been classified earlier and on the number of flows that have an impact on the current classification. In particular, the reason resides in the sequence of previous flows. We can anticipate that this behavior depends on the wrong classification of POP3 and HTTP traffic (see Fig. 5 and Fig. 9) and we will discuss it more in the following.

A value of  $\lambda = 0.9$  gives the best performance for the classifier. We obtain a precision of 94% already after two packets, 97% after four packets and 99.9% when ten packets are used for the classification. When  $\lambda = 0.1$  the classifier predicts will less accuracy the applications. With this value of  $\lambda$  the classifier is more sensitive to recent flows. Thus, it is more prone to a wrong classification when the host has a uniform traffic behavior over all applications or the latest flows are wrongly classified in case of when the host does a unique type of traffic.

For all selection of  $\lambda$  we have better performance compare to a classification without using the host profile information ( $\lambda = 1$ ). 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 that is browsing the web is more prone to have a sequence of HTTP connections.

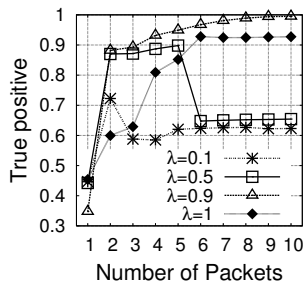


Fig. 5: TP ratio - POP3

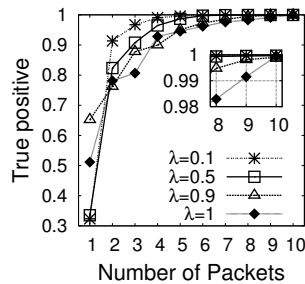


Fig. 6: TP ratio - HTTP

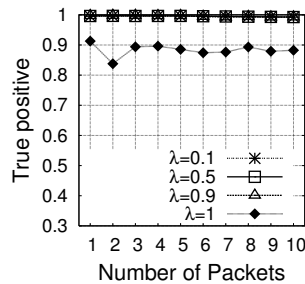


Fig. 7: TP ratio - SMTP

**True Positive** Fig. 5 shows the True Positive (TP) ratio for the POP3 application as a function of the number of packets used for the classification. We can see that for  $\lambda = 0.9$  the TP ratio increases when we use more packets for the classification, until it reaches a value of 0.99 with ten packets. In comparison,

when the host information is not taken into account, the TP ratio reaches a maximum value of 0.91 with 10 packets. As we have anticipated above, the TP ratio drops when  $\lambda = 0.5$  and 6 packets are used for the classification. Similar performance can be observed for  $\lambda = 0.1$  at the second packet. This means that the classifier fails to identify correctly POP3 flows as they are assigned to other applications. We will confirm this behavior when we analyze the False Positive ratio to understand what is the output application of our classification method. The fact that this happens only for these small values of  $\lambda$  means that the output of the classifier before considering the host profile of a user is uncertain such that the result changes in favor of the application recently classified for the same host.

Fig. 6 and Fig. 7 plot the True Positive (TP) ratio for HTTP and SMTP as a function of the number of packets respectively. Fig. 6 shows that the TP ratio increases for all the values of  $\lambda$ , even when we do not use any host information. The classifier has better performance with  $\lambda = 0.1$ , which means that there are consecutive HTTP flows in general. From Fig. 7 it is interesting to notice that the TP ratio is 1 for any number of packets of SMTP classified traffic when we use the profile of the host to refine the classification. The fact that the result is not sensitive to the value of  $\lambda$  means that the SMTP traffic is predominant in some hosts. We have also already observed in Fig. 2 and Fig. 3 that the size of the SMTP traffic has a different distribution than the other applications, which makes easy the classification of SMTP traffic. Besides, when packets of other applications have a size which is recognized as a SMTP signature then the flow is also classified as SMTP. However to confirm this explanation we need to discuss the false positive ratio.

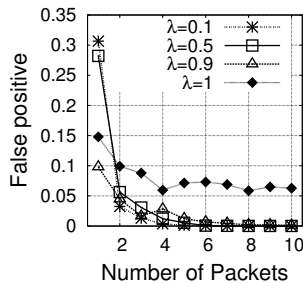


Fig. 8: FP ratio - POP3

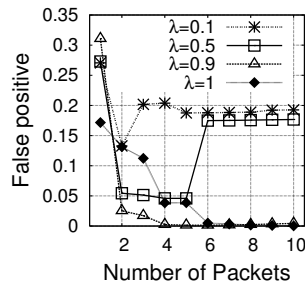


Fig. 9: FP ratio - HTTP

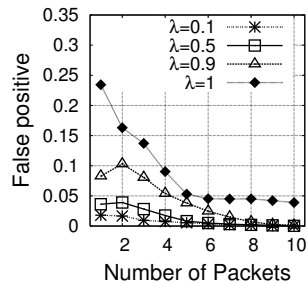


Fig. 10: FP ratio - SMTP

**False Positive** Finally we discuss the results of the False Positive ratio to confirm our previous hypotheses. We can immediately notice in Fig. 8 that the percentage of misclassified flows of other application, assigned to POP3, drops significantly after 4 packets. Thus, the True Positive ratio of 5 indicates the correctly classified POP3 flows. Fig. 9 also confirms that most of the POP3 flows that have not been detected are indeed classified as HTTP traffic. This is shown for values of  $\lambda = 0.1$  and  $\lambda = 0.5$ . We also observe that when  $\lambda = 0.9$  the false positive drops to 0 after four packets only in Fig. 8 and Fig. 6.

Fig. 10 shows that the classification for most of the SMTP traffic is indeed correct when the classification of recent flows weigh more. When  $\lambda = 0.9$ , the

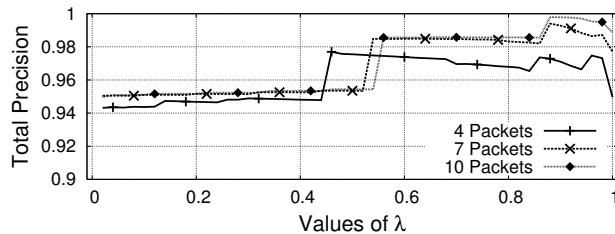


Fig. 11: Total precision computed after 4, 7, and 10 packets as a function of  $\lambda$

classifier labels other flows as SMTP, which means that some hosts have SMTP flows that interleave the ones of other applications. In the next sections we discuss the impact of  $\lambda$  for the classification. Then we have analyzed the profile of the hosts to understand which services the hosts run and to determine their traffic pattern.

### 5.3 Importance of the discounting factor $\lambda$

Fig. 11 plots the total precision as a function of  $\lambda$ . The discounting factor determines how previous flows are considered for the classification of a new one. We recall that for  $\lambda$  close to 1 the host profile is computed over a longer period and previous flows have similar weights. The opposite case is when  $\lambda$  is close to 0. Finally  $\lambda = 1$  means that the classification does not account for the profile of the host.

A precision above 0.9 for all values of  $\lambda$  shows that our iterative method is effective in classifying the applications. However, we can see that the precision is a function of  $\lambda$ . In particular we notice that when few packets are used for the classification (4 in the plot), the profile of the user helps for an early detection of the Internet traffic. In this case, the machine learning algorithm does not have much information about a flow and it associates to the applications comparable probabilities. Thus, the profile of the host has higher influence on the final classification.

However, a correct  $\lambda$  is important to improve the classification even if more information is available. From Fig. 11, we can notice that high values perform better. The host profile prediction of the application is more accurate because we associate a probability to the application of the next flow based on the fraction of the past traffic due to this application. In the next section, we characterize this traffic and how the flows are distributed.

### 5.4 Traffic pattern of a host

In the traces, described in Table 2, we have identified different types of hosts. In particular, there are some hosts within the Brescia campus that are dedicated to single services and other hosts that use all the three applications. In this section, we analyze in details the traffic of the latter type of host and we determine its profile. This will shed light on the importance of  $\lambda$  for the identification of the

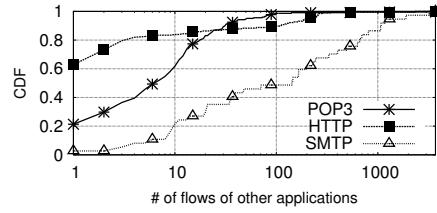
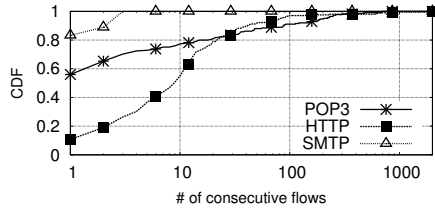


Fig. 12: CDF of the number of consecutive flows of the applications for IP1      Fig. 13: CDF of the number of flows of other applications for IP1

Internet traffic. The total number of flows generated by this IP, noted *IP1*, is 14,151 flows, which is subdivided as follows: 7,101 HTTP flows, 7,014 POP3 flows, and only 35 SMTP flows.

Fig. 12 plots the cumulative distribution (CDF) of consecutive flows of the same type of application in semi-log scale. Since there are only 35 SMTP flows, most of these flows are isolated so that there are only sequences of 1 to 3 consecutive flows (burst). Almost 60% of the POP3 traffic consists of a single flow. The rest of the POP3 traffic consists of single larger bursts and the largest one consists of almost 700 flows. The number of small HTTP consecutive flows is 10% of the total number of HTTP bursts and 80% of the burst consists of less than 100 flows. This corresponds to a typical browsing activity of a user, who surfs from one web site to another by following the hyperlinks.

The fact that the medium size burst of HTTP traffic are interleaved with POP3 is evident when we use a small value of  $\lambda$ . In particular, since there are more consecutive HTTP flows than POP3, the host based classification with a small value of  $\lambda$  is more sensitive to the presence of a new burst. Moreover, the POP3 and HTTP packet size has similar distribution for certain packet numbers, as discussed in Section 5.1, thus the host profile has more impact in the classification decision.

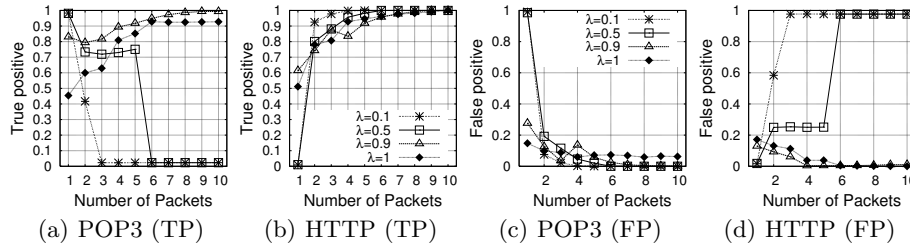


Fig. 14: Classification results for the analyzed host IP1

To conclude the characterization of the profile, Fig. 13 plots the cumulative distribution (CDF) of flows of other applications that separate two flows of the same application. There are more than 60% of HTTP bursts separated by only 1 flow of another application and 80% of HTTP bursts separated by less than 10 consecutive flows. As for POP3, there are 20% and 60% of burst respectively. The presence of small burst of other applications between two HTTP flows justifies the high percentage of false positives (see Fig. 9). In general this is more

pronounced with small values of  $\lambda$ . In Fig. 14 we show the classification results for *IP1*. The results confirm the analysis of the traces. The impact of the profile of the host is also evident.

## 6 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 a real trace and we profile the hosts with the same IP prefix. We test our method for different values of the discounting factor  $\lambda$  and discuss the optimal choice based on the traffic pattern of the host. 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 0.99. Finally, we characterize the host profile and show the distribution of the flows, i.e., the traffic pattern of a representative host.

## References

1. IANA: Internet assigned numbers authority. <http://www.iana.org/>
2. Moore, A., Papagiannaki, K.: Toward the accurate identification of network applications. In: Proceedings of the 6th Passive and Active Measurement Workshop (PAM 2005). (October 2005) pages 41–54
3. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: WWW 2004 Conference, Philadelphia, USA (May 2004)
4. Moore, A., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: ACM Sigmetrics. (2005)
5. Mcgregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow clustering using machine learning techniques. In: PAM. (2004) 205–214
6. Bernaille, L., Teixeira, R., Salamatian, K.: Early application identification. In: The 2nd ADETTI/ISCTE CoNEXT Conference, Lisboa, Portugal (December 2006)
7. Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L.: Traffic classification through simple statistical fingerprinting. In: ACM-Sigcomm Computer Communication Review. Volume 37. (January 2007) 5–16
8. Jaber, M., Barakat, C.: Enhancing application identification by means of sequential testing. In: NETWORKING '09: Proceedings of the 8th International IFIP-TC 6 Networking Conference, Berlin, Heidelberg, Springer-Verlag (2009) 287–300
9. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: Blinc: Multilevel traffic classification in the dark. In: SIGCOMM'05, New York, USA (August 2005)
10. Trestian, I., Ranjan, S., Kuzmanovi, A., Nucci, A.: Unconstrained endpoint profiling (googling the internet). In: ACM SIGCOMM. SIGCOMM '08, Seattle, WA, USA, ACM (2008) 279–290