A popularity-aware method for discovering server IP addresses related to websites

Luis Miguel Torres, Eduardo Magaña, Mikel Izal and Daniel Morato
Departamento de Automática y Computación, Universidad Pública de Navarra, Pamplona, Spain.
Email: [luismiguel.torres, eduardo.magana, mikel.izal, daniel.morato]@unavarra.es

***Abstract*—The complexity of web traffic has grown in the past years as websites evolve and new services are provided over the HTTP protocol. When accessing a website, multiple connections to different servers are opened and it is usually difficult to distinguish which servers are related to which sites. However, this information is useful from the perspective of security and accounting and can also help to label web traffic and use it as ground truth for traffic classification systems. In this paper we present a method to discover server IP addresses related to specific websites in a traffic trace. Our method uses NetFlow-type records which makes it scalable and impervious to encryption of packet payloads. It is, moreover, popularity-aware in the sense that it takes into consideration the differences in the number of accesses to each site in order to provide a better identification of servers. The method can be used to gather data from a group of interesting websites or, by applying it to a representative set of websites, it can label a sizeable number of connections in a packet trace.**

I. Introduction

The web is probably the classic Internet application that has grown and changed the most during the past two decades. The simple and mostly static webpages of the 1990s have given way to interactive sites with dynamic content that may come from different providers and be tailored to the specific characteristics of the users. This evolution has made common the use of web analytic tools, content distribution networks (CDN) or client-side processing. Moreover, web traffic, besides traditional web browsing, is now associated with a myriad of other services from webmail to video-streaming and online games. These dramatic changes have affected the characteristics of web traffic and recent studies [1] show that they are indeed different than the (simpler) ones described thoroughly in the past [2].

The evolution of the web mirrors the changes in the Internet traffic as a whole. With the appearance of new applications, work was done in order to develop techniques that were able to distinguish between their traffic. As a consequence, if we want to label the traffic of different applications, some techniques exist from port mapping to signature-based and behaviour-based classification. However, nowadays it is not enough to know that a particular TCP connection (or *flow*) carries web traffic. From a network administrator point of view it is very interesting to know which is the actual service being provided through the web application. This is a complex and less studied problem, the solution of which presents immediate applications. First of all, as we stated previously, this information is very interesting from the perspective of security and accounting. Additionally, labelling the connections of specific websites and services will allow a thorough study and modelling of their traffic. Finally, the resulting labelled traces can be used as ground truth in order to tune or test new traffic classification systems.

All in all, the problem we present is to label traffic from just one application (the web) depending on the session (connection to a website) during which it was generated. This labelling process is far from trivial. Manually labelling a complete trace with multiple users accessing a variety of websites during a long period of time is infeasible. The number of connections is too high and a big part of them will prove difficult if not impossible to label even if we were to do it one by one.

We have chosen to work at flow level for simplicity and scalability. We use NetFlow-type records [3] in which just the basic information (timestamps, IP addresses, ports, size) is stored for each bidirectional flow. This summarized information is easier to capture and process, even in real time and it is not affected by the encryption of the packet payloads.

For the captured flows, we consider each client IP address, which belongs to our network, as a *user*. We make the assumption that the server IP addresses of those flows can be mapped to a website. For some of them, this will not be true: websites may share a server provided by a common hosting service or may even share content from a third party server. However, a sizeable number of addresses should be related to just one website, at least during a relatively short period of time. We will, then, try to label not each TCP connection but groups of connections to the same server IP addresses.

We define a *session* as a collection of TCP flows generated by the web browser while the user is accessing a specific website. For example, a session to a webmail site would ideally span all the connections opened by the web browser from the moment the user opened the login webpage of the mail service until he or she closes the browser, the tab or opens a different website in the same tab. However, the beginning and ending of a session are difficult to infer from the captured traffic. In order to do so, we further simplify by limiting the number of websites whose connections we want to find and label. At the same time of the web traffic capture, we also collect DNS traffic from our DNS server. We consider the most popular IP addresses which are associated to second level domains. We will use their apparition in the trace as a signal of the beginning of a session to the related website.

By studying different sessions from different users to the

same sites, we will be able to obtain a list of IP addresses related to the sites. In the following sections we will present a method in order to do so in a reliable way. With this methodology we will be able to label the traffic of interesting websites for subsequent study. Moreover, we believe that by selecting an appropriated number of popular websites we can obtain a sufficient set of labelled connections that can be used to tune and test other classification systems.

II. Related Work

Network traffic classification is a widely studied field. Traditionally the objective has been to classify the traffic depending on the application that generated it (e.g. web, file transfer, e-mail, etc.) The earliest techniques relied on simple port-based classification but this proved unreliable prompting the development of new methods [4]. Signature-based techniques are widely used today [5] but the appearance of a sizeable number of new and rapidly changing applications and the increase in the use of encryption has inspired classification techniques based on the statistical characteristics of the traffic [6].

When it comes to identifying different services provided by an application (in our case, the web) there are less precedents. Some work has been done in characterizing the traffic of certain services like Youtube [7], [8]. However, most studies focus on the social characteristics (popularity of videos, etc.) or study the effect of the video codecs in the per packet statistics of the traffic generated. Other studies have compared traditional HTTP traffic with new services like social networks [9] or interactive AJAX-based services [10].

Some efforts directed to classifying the traffic of the different services provided through HTTP have started appearing in the last years. Schatzmann *et al.* [11] try to design a method able to distinguish webmail flows from other HTTPS connections using NetFlow records. Archibald *et al.* [12] propose a technique that uses flow level statistics to classify the traffic of three different services represented by Facebook (social network), Gmail (webmail) and Youtube (video streaming). In any case, these new classification proposals have been inspired by the application classification techniques we mentioned previously. They usually rely on machine learning schemes (supervised classification or clustering) which need labelled data sets for tuning and testing. The authors usually stress the labelling process as difficult and time consuming and there is where the relevance of the work presented in this paper resides.

A different approach is presented in [13] where the authors use DNS information to "untangle" web traffic. They introduce DN-Hunter, a tool able to relate traffic flows with content providers on the fly by analysing DNS responses. It is an interesting concept that yields good results but it requires capturing all DNS traffic directed to the individual users (something that, for example in our case was not possible) and can only relate a particular server to a website if there is a relationship between them in the DNS information.

TABLE I
Traffic traces

Trace	Date of capture	# Flows	# Users	# Sel. websites
Trace 1	Jan 14 - 23, 2013	11M	1096	66
Trace 2	Apr 5 - 19, 2013	16M	967	73

In our case we have decided to tackle the labelling of web traffic traces by centering the study in basic flow information (timestamps and IP addresses). In [14] we presented a simple system able to cluster TCP flows into web sessions on the fly using time and server IP address proximity. The difficulty in tuning and testing that system prompted the study we presented in [15]. This paper builds over that proposal improving the assignement method by taking into account the popularity of the websites and introducing a complementary method based on IP subnetworks.

III. Scenario and datasets

A. Network scenario

For the different analysis presented in this paper, we use two traffic traces captured in the Internet link of the Public University of Navarre. In order to obtain flow records rather than the usual packet traces, we use Argus. Argus [16] is an open-source audit tool that is able to generate flow reports with the same features (and more) than NetFlow/IPFIX. Basic information about the traces is shown in Table I. As we are only interested in (outbound) web traffic we filtered all non-TCP connections and those TCP connections whose destination port was not 80 or 443 as it is widely assumed that they represent the majority of HTTP traffic. We also eliminate flows from IP addresses that we know are NAT routers so we can make the assumption that each IP address we see from our network represents a single user. With this, we obtain two data sets in which, for each flow, the following data is stored: initial and final timestamps, source and destination IP addresses, source and destination TCP ports and total number of packets and bytes. This greatly reduces the initial volume of data, making it more manageable.

In addition to the flow records, we captured all DNS traffic between our DNS server and the Internet (our vantage point did not allow the capture of DNS traffic from the users to the DNS server as in [13]). In this case we capture full packets in pcap format as we will extract information from fields in the DNS payload. This DNS traffic is not necessary for the labelling process we present in the paper but we will use it for tuning and validating our system and for choosing the list of relevant websites as explained in the following subsection.

B. Website selection

As stated in section I, we seek to identify IP addresses related to a predefined set of websites. If we want to label as many IP addresses as possible, it will make sense to select the most popular sites in the trace (i.e. the sites with more sessions). In order to do so we will follow these steps:

- We separate the flows in the trace by user and, considering their start times, in intervals of 120s (in [15] we selected 120s as a good higher threshold for session length).
- A server IP address will be popular depending on how many of these intervals it appears in. Defining popularity like this instead of just considering the number of flows for each IP address minimizes the effect of websites which open multiple connections to the same addresses during a session. In those cases, an IP address could have a big number of flows with a small number of sessions for the associated website.
- Following this definition, we have selected the 2,000 most popular server IP addresses in each trace. Of these popular addresses we consider the ones that, in the DNS capture, have an associated domain name in the form of xxx.xxx or www.xxx.xxx. Websites with the same second level domain name are grouped together (e.g. twitter.com, twitter.es and www.twitter.com).
- We manually select the interesting websites filtering advertising servers or web tracking services.

With this procedure we have obtained a list of 66 sites for trace 1, some of them worldwide known and others which are popular in our local community (e.g. Tuenti, a Spanish social network or a number of local newspapers). For trace 2 we obtained a list of 73 websites. Both traces share the same very popular websites although there are some differences in the less popular ones. Each of the selected websites has one (or more) associated IP address that we will use in order to identify their sessions. Moreover, as these IP addresses are associated to the "main" domain name of the site (i.e. www.facebook.com rather than, for example, s-static.ak.facebook.com), we expect that connections to them will happen at the beginning of the sessions of the website.

A similar list of IP addresses can also be obtained from a predefined list of websites by making automatic DNS queries for their domain names during the time span of the traffic capture. This can be used to label the server IP addresses of specific websites rather than the most popular ones. In that case it is not necessary to capture any DNS traffic (aside from the responses to these automatic queries). We used that approach in [15] where we observed that the server IP addresses associated to "main" domain names usually remain the same during the span of a few days/weeks.

IV. Popularity-aware labelling

Modern websites present dynamic content that may be stored in various different servers. Some of these servers may even provide content for different websites as it can be the case with CDNs. As a consequence, the IP addresses that are accessed when loading a website change over time and some of them may be used by more than one site. Nevertheless, intuitively, if we capture enough sessions to the same websites, a number of IP addresses are bound to start appearing repeatedly. Also intuitively, these IP addresses must belong to servers that store the fundamental content of the site as opposed to some images, videos, advertisements and other rapidly-changing or third party content that can be stored in servers which appear only occasionally in the sessions of a website (or appear in sessions of multiple websites).

The method we present in this section takes all this into consideration in order to achieve a reliable labelling of server IP addresses in the trace. It can be divided in three steps: finding web sessions, labelling candidate IP addresses by concentration method and labelling candidate IP addresses by subnetworks method. In the three steps, some necessary parameters will be left as variables in order to be tuned with experimental data.

A. Finding web sessions

We have defined a web session as the set of connections generated by the web browser while the user is accessing a specific website. It should be noted that, even if we know when a user is accessing a website, we have no way to assess if all the connections truly are caused by the load of that website. Users may use other applications that use the ports normally associated to HTTP(S) and may open concurrent sessions to other websites (a frequent happening given the widespread tab-based design of web browsers).

Concurrent sessions are, in fact, one of our main concerns. Even though we are considering the most popular websites in our network, the differences in number of sessions are very big (e.g. more than three orders of magnitude between the most popular site, Google, and the 50th). Because of this, if we consider a very popular site against one of the less accessed, it is possible that most of the sessions of the latter happen during sessions of the former. In this case it would be difficult to assign correctly the IP addresses of the less popular site as they also always appear in sessions of the other one. Moreover, if we consider the less popular websites that have not made it into our list (for the sake of brevity we will call them *unknown websites* from now on), we have no way of knowing if there is a concurrent session to one of them at any given moment so their IP addresses could be assigned incorrectly.

Taking this into consideration we have modified the way we defined the sessions in [15] introducing measures that help the identification of these IP addresses. Sessions in a trace are created as follows:

- We use the popular IP addresses described in III-B as signals of the beginning of a session. We will call them *main IP addresses*. In other words, when we find a connection from a user to one of the main IP addresses, we consider that the user is visiting the corresponding website.
- The following connections initiated by that user during a period of time (*session length*) will be considered part of the same session. Their server IP addresses will be then *candidate IP addresses* that, in the end, may or may not be associated with the website.
- If the user accesses a main IP address during a session of a different website a new session will start. The system takes now into account the popularity of both sites. If

the new session is associated to a more popular site, the following connections will still be assigned to the older session until it finishes. On the other hand, if the new session is associated to a less popular site, the following connections will be assigned to it, except for those whose server IP are already candidate IP addresses in the older session.

- If there are no active sessions and the user opens a connection to an unknown (not main) IP address, a session to an unknown site is created. If a session to a known site starts while this session is active, all new connections will be assigned to the new session except those whose server IP addresses are already assigned to the unknown session. The only purpose of this unknown session is to protect the addresses of the sites we are not considering from being incorrectly assigned to known sites.

We have not specified when a session ends as choosing an indicator for session ending is not so simple. Sessions are variable in length depending on the type of service accessed and on user behaviour. Although in section III-B we used 120s as an approximate value, we have chosen to leave session length as a variable parameter that can be tuned in order to obtain the best results. We will discuss this in section V.

B. Labelling by concentration of IP appearances

By applying the previous step to a traffic trace we obtain a set of sessions. Each session is related to a user which, as defined, is the source IP address in every connection, and to a website depending on the main IP address that originated the session. By considering all the sessions related to a website, we gather a list of candidate IP addresses for that website.

As a first step, for each website, we only will attempt to label candidate IP addresses that appear in two different sessions of two different users. This allows us to eliminate a big number of candidate IP addresses that are not strongly related to the website or may appear in its sessions because of the specific behaviour of a single user. In order to make this requirement fair for all sites we take into account that, of the sites we are considering, the least popular ones have around 200 sessions in the trace. This means that we are only considering IP addresses that appear in more than 1% of the sessions of these websites. We will add this requirement for the candidate IP addresses of the more popular sites so all candidate IP addresses are on equal ground.

Of the resulting list of candidate IP addresses, some appear only in sessions of one of the websites under study while others appear in more than one. It is clear that, at this stage, we cannot assign the latter IP addresses to any website. But we cannot do it either for the candidate IP addresses that appear only in sessions of one website. Some of them will be related to that website but others may belong to unknown websites. In order to ensure that an IP address is related to a website we use the concentration of IP appearances as a decision parameter.

We define the *concentration of IP appearances* for a candidate IP address of a website as the ratio between the number of appearances (i.e. TCP flows) of the candidate IP in sessions of the website and the total number of appearances of the IP in the trace. It is important to notice that connections to a candidate IP may be opened outside the corresponding website sessions. For one thing, sessions, as they have been defined, may not encompass all the actual connections. Furthermore, a website may be accessed without a previous connection to a corresponding main IP when, for example, following a hyperlink. Nevertheless, a high value of this ratio strongly suggests that the candidate IP is related to the website.

As a consequence, for all the candidate IP addresses of a website, we will label those whose concentration of IP appearances is higher than a given threshold. We want to set a value for this threshold that is high enough to avoid erroneous labelling but not so high that most candidate IP addresses are left unlabelled. As with the session length, we will adjust this parameter for the best results in section V.

C. Labelling by subnetworks

If we study the candidate IP addresses for each website it becomes clear that a sizeable number of them belong to the same IP subnetworks. This is something to be expected as the websites host part of their content in servers that belong to the same company and the IP addressing space that each company uses is limited. The concentration of IP appearances is good for discovering individual servers related to a website but we hypothesise that we could use this network relationships between candidate IP addresses in order to increase the number of them we are able to label.

Aside from main IP addresses, in [15] we found that few IP addresses appear in most sessions of a website. Small websites usually host a lot of their content (photos, videos, etc.) in different third party servers. Very popular websites, on the other hand, usually have a big server infrastructure and content may come from different IP addresses depending on their internal politics. However, if we consider IP subnetworks rather than individual IP addresses, we will find that it is more common to find connections to these networks in most sessions of a website. We have selected subnetworks of class-C size as we believe it is a size small enough so that servers from different websites rarely appear in the same subnetwork.

The labelling process in this case will follow these steps:

- We group the candidate IP addresses of a website in IP subnetworks. In this case, we do not eliminate candidate IP addresses that do not appear in, at least, 1% of the sessions of each website as the filtering will come after, when we consider the whole subnetworks. We only consider a class-C subnetwork if there are connections to, at least, two different server IP addresses that belong to said network in sessions of the site.
- We calculate in how many sessions of the website a connection to an IP from the subnetwork appears.
- We filter the candidate subnetworks according to a *subnetworks session threshold*. We will assign a subnetwork to a website if connections to it appear in a percentage of its sessions that is higher than the threshold.

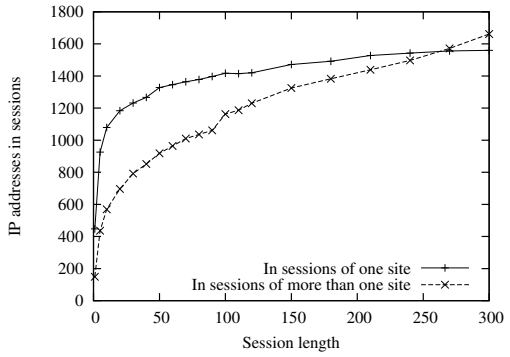

Fig. 1. Candidate IP addresses in sessions

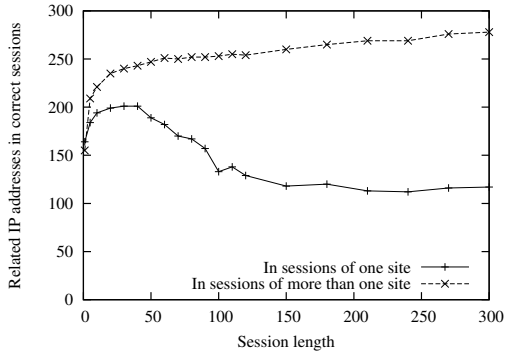

Fig. 2. Related IP addresses in sessions

- If a subnetwork has been assigned to only one website, we will label the IP addresses in that network as belonging to the website.
- If a subnetwork has been assigned to more than one website we will assign only the IP addresses of the subnetwork that appear in only one website.

A subnetwork will be assigned to more than one website primarily because of two reasons: it is a CDN subnetwork or it is a subnetwork of a popular site (i.e. Google) the connections of which usually appear in sessions of less popular websites. As we cannot distinguish between these two situations we only assign the IP addresses that appear in just one website. If the network belongs to a popular website we will only assign some of the IP addresses but we will not assign them incorrectly to less popular websites. If the network belongs to a CDN, we will only assign IP addresses of servers in the CDN that host content of a specific website. Again, the value of the subnetworks session threshold will be selected in the tuning section.

V. Tuning

In the previous section we presented our system but we left some parameters without value as we wanted to tune them with experimental data to obtain the best possible results. In this section we will tune these parameters using the data from trace 1. Because of this, all figures in this section present data from that trace. In section VI we will check if the selected values also yield good results with trace 2.

A. Choosing web session length

The first parameter we are going to tune is session length. A big value for this parameter will result in long website sessions. Intuitively this will allow a better labelling of the IP addresses of websites that usually are related to long sessions (like, for example, online newspapers). However, there is a drawback: as we increase the value of session length, the probability of overlapping sessions of different websites also increases. Due to the popularity-aware nature of our method, this is not a huge problem for our considered websites as the sessions of the least popular ones are protected from overlapping while the most popular will appear elsewhere in the trace anyway. Nevertheless, our method can do little to protect the IP addresses of unknown websites and if we use very big sessions we are bound to make some mistakes with them. Moreover, longer session lengths imply higher processing requirements as more information must be kept in memory during the execution of our labelling system.

Taking all this into account we first consider Figure 1. In this Figure we represent, for different session lengths, the number of candidate IP addresses that appear in all the sessions to all the considered websites combined. We are not applying any of the labelling thresholds yet. The continuous line represents IP addresses that appear only in sessions of one website; the dashed one, IP addresses that appear in sessions of more than one website. It seems interesting to get a big number of IP addresses that appear only in sessions of one website as they are good candidates for labelling. However, as we can see, that curve ceases to grow for small values of session length. On the other hand, the growth of the number of IP addresses that appear in sessions of various websites suggest, as predicted, that increasing the length of the sessions results in more overlapping between websites. It is important to note, however, that even if an IP address appears in sessions of more than one website, it may appear in a lot of sessions of one of them and in only a few of the others. The concentration of IP appearances parameter will then be able to assign it to the correct site.

Figure 1 is interesting in that it shows that a high value for session length is unnecessary. However, the fact that an IP address appears only in sessions of one of the considered websites does not imply that this address truly belongs to the website. In Figure 2 we consider a subset of the candidate IP addresses that we will call *DNS-related IP addresses*. An IP address will be DNS-related to a website if it has an associated domain name that contains the name of the website (e.g. an IP that appeared in a DNS response for profile.ak.facebook.com.edgesuite.net will be DNS-related to facebook). Not all IP addresses that belong to a website will be DNS-related (companies have different naming policies for their servers and some websites may host some of their content in third-party servers). In this Figure, the dashed line represents the number of DNS-related IP addresses that appear in sessions of the correct website. Of those, the ones that appear **only** in sessions of the correct website are represented

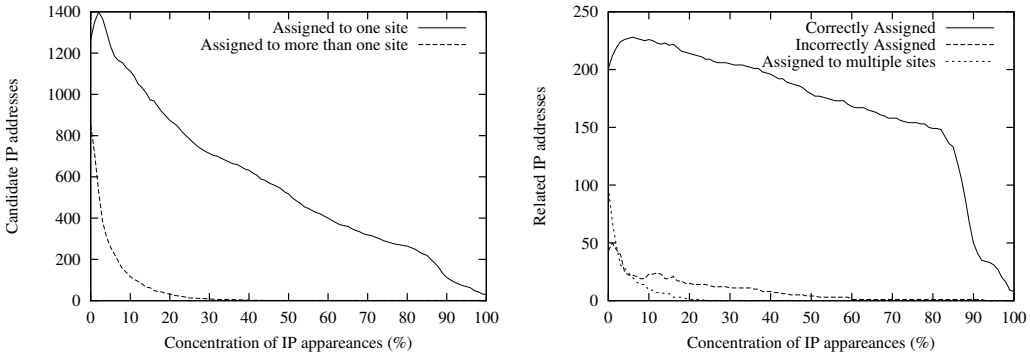

Fig. 3. Concentration method: (A) Assigned candidate IP addresses; (B) DNS-related IP addresses

by the continuous line. Again, increasing session length past 50 seconds does little in order to increase the number of DNS-related IP addresses that appear in sessions of their websites. Also, the number of related IP addresses that appear only in sessions of the correct website decreases after reaching a maximum in 40 seconds. In view of these results, we choose 40 seconds as the value for session length.

B. Choosing a concentration threshold

With 40 seconds as session length and filtering the IP addresses that do not appear in, at least, two sessions (or 1% of the total sessions of the website) of two different users, we obtain a total of 2,119 candidate IP addresses for the 66 websites combined. Of these, 1,267 appear only in sessions of a website and 852 appear in sessions of more than one website. The fact that some candidate IP addresses appear in sessions of more than one website suggests that some of the candidate IP addresses that appear only in one site may actually appear also in sessions of unknown sites. Because of this, if we want the concentration threshold to be able to identify the candidate IP addresses that truly belong to a site, the chosen value should, in the first place, ensure that no candidate IP address is assigned to more than one of the considered websites.

As we see in figure 3.A the parameter works well in this respect. In this figure we represent, for different values of the concentration of IP appearances threshold, the assigned candidate IP addresses for all the websites. The continuous line represents the candidate IP addresses assigned to only one site and the dashed one, the candidate IP addresses assigned to multiple sites. Increasing the threshold produces a sharp decrease of the latter that nears zero for values higher than 30%.

However, as it happened with session length we do not know if the candidate IP addresses are being correctly or incorrectly assigned. For figure 3.B we consider DNS-related IP addresses again. The continuous line represents the number of DNS-related IP addresses correctly assigned to the websites; the long-dashed one, the number of incorrectly assigned DNS-related IP addresses; and the short-dashed one, the DNS-related addresses assigned to multiple sites. Addresses assigned to multiple sites do not suppose a problem as they disappear quickly. Moreover, as we increment the threshold, the number of addresses assigned wrongly also decreases. For values of the concentration of IP appareances parameter over 40% the error ratio is less than 5% and therefore we will choose this value for the labelling threshold.

C. Choosing a subnetworks threshold

Analysing the 40 second sessions, we obtain 565 class-C subnetworks of which 246 appear in sessions of only one website. Figure 4 is analogous to the one presented in the previous subsection. As we can see, the number of IP addresses assigned to various websites is higher in this case and increasing the subnetworks threshold does not lower it much. As we predicted, most of these IP addresses belong either to very popular websites (especially Google) or to CDNs. However, the number of DNS-related IP addresses assigned wrongly is low (lower than 5% for values of the threshold above 50%). We will set 50% as the value of the subnetwork sessions threshold. Another reason to choose a 50% threshold is the sharp decrease in the number of assigned IP addresses that happens between 50% and 60%. In fact, at 50% 95 subnetworks are assigned to the sites but 29 of them appear in less than 60% of the sessions of their websites and will disappear if we increase the threshold. The fact that there is a sharp decrease both in the total assigned IP addresses and the DNS-related IP addresses suggest that these networks were correctly assigned.

VI. Results and discussion

Once we have selected the values of the different thresholds we test our system with traces 1 and 2. We want to check if once the tuning is performed for our network with one of the traces, the results will still be good for the other. In Table II we show a breakdown of the assignation results for both traces. The first columns show the individual results of both methods. Then we show the combined number of assigned IP addresses, how many of them were assigned to the same website by both methods (agreements) and how many were assigned differently (disagreements).

The behaviour of the labelling system seems consistent enough for the two data sets. The number of labelled IP

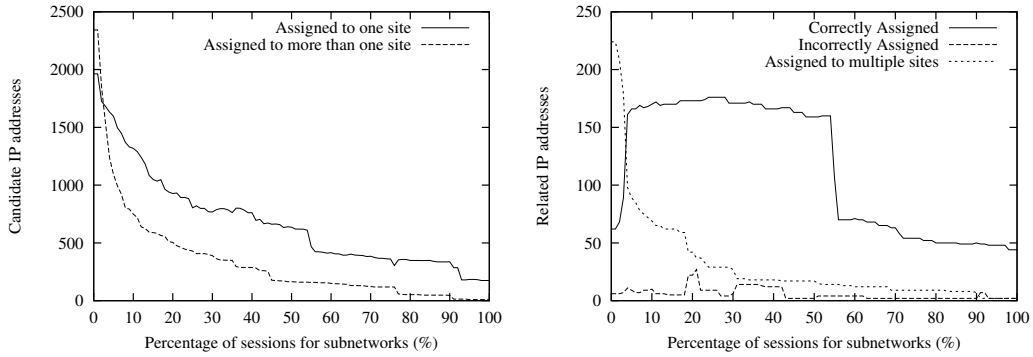

Fig. 4. Subnetworks method: (A) Assigned candidate IP addresses; (B) DNS-related IP addresses

TABLE II
ASSIGNATION RESULTS

Traffic Trace	C. method Assigned	S. method Assigned	Both methods Assigned	Agree	Disagree
Trace 1	632	636	1268	182	6
Trace 2	574	591	1140	103	5

addresses is similar for the two methods which are complementary. As only 9-15% of the resulting IP addresses are shared by the two methods, applying both of them allows labelling a much bigger number of connections. Addresses labelled differently are a rare occurrence (around 0.5% in the worst case) which is a promising indicator of the precision of the system.

A. Validation

Validating the obtained results is a challenging process. Identifying the assigned IP addresses manually is a time consuming task that may prove to be impossible in some cases as the tools we can use are limited:

- Simply trying to access the web server (e.g. by typing the IP address in a web browser address bar) is not useful in most of the cases as servers expect to be asked for specific content and will provide a standard error page or simply reject the connection.
- Studying the application data of the packets of the connections to the IP address may sometimes help. In our case, the sniffer that captures the Internet traffic of our University limits the capture size to 100 bytes per packet. Because of that, we rarely see past the HTTP GET field and we have not found it very useful for identification purposes.
- Some IP addresses may be identified by information gathered via the WHOIS protocol. However, this only works if we can draw a relationship between the owner of the IP and the actual website.
- In the end, the most interesting and easy to access source of information is DNS and that is why we have used it in order to gather the DNS-related IP addresses in past sections.

In Table III we show the results of the assignation for the DNS-related IP addresses. For both methods we present: the number of DNS-related IP addresses assigned (A. IPs), the ratio of the DNS-related IP addresses against the total assigned IP addresses, and the precision in IP addresses, flows and bytes (P.IP, P.Flow and P.Byte). We define the *precision* as the ratio between the number of true positives (correctly labelled addresses) and the total number of assignations. For example, a precision ratio of 95% for IP addresses means that out of 100 DNS-related IP addresses, 95 were assigned to the correct website. Taking into account the number of flows and the bytes in those flows for each IP we can also obtain the flow and byte precisions.

TABLE III
ASSIGNATION RESULTS FOR DNS-RELATED IP ADDRESSES

Traffic Trace	Concentration method A. IPs	of total	P.IP	P.Flow	P.Byte
Trace 1	197	31%	95.9%	94.0%	99.5%
Trace 2	183	32%	93.4%	90.7%	98.7%
Traffic Trace	**Subnetwork method** A. IPs	of total	P.IP	P.Flow	P.Byte
Trace 1	159	25%	98.7%	99.0%	99.9%
Trace 2	173	29%	92.4%	86.2%	97.9%

The assignation results are generally good. For trace 1 they are better as the tuning of the system was made in order to minimize the wrong assignations in that trace. However, all values of precision remain near 90% for trace 2. It must be noted that, although trace 2 was captured in the same network and under the same conditions, it was captured months after trace 1, it is longer and the selected websites for it are not the same. Even with these differences, the performance of the system for the DNS-related IP addresses remains acceptable.

Given that a representative amount (25%-32%) of the assigned IP addresses are DNS-related IP addresses it is tempting to extended these results for all the assigned IP addresses. However, there are two main concerns: (i) the DNS-related IP addresses are not selected randomly from the total assigned IP addresses so their behaviour may not be extensible for all of them. (ii) the DNS-related IP addresses only consider

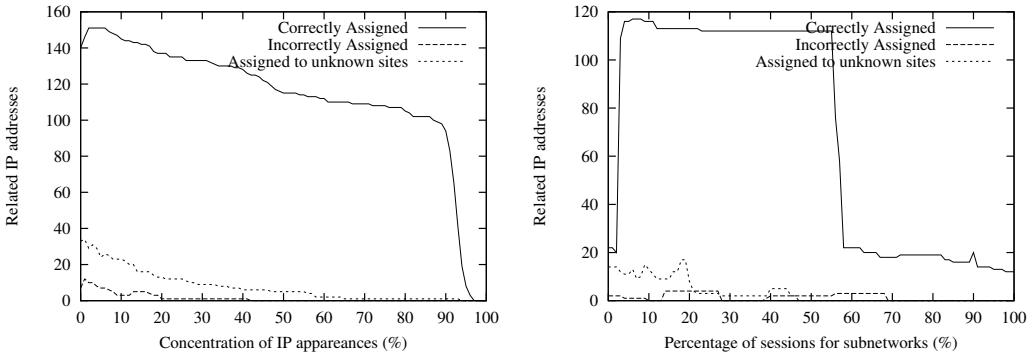

Fig. 5. Assigned DNS-related IP addresses with unknown sites for concentration (A) and subnetworks (B) methods

the preselected websites so we have no way of proving that we are not assigning IP addresses from unknown websites wrongly. Both concerns are related as the possible mistakes in the assignation of non DNS-related IP addresses primarily affect unknown sites.

Figure 5 addresses this problem. The figures are the same as 3.B and 4.B but we have taken half of the 66 considered sites out of the labelling system and used them as a control group. We have taken both very popular and less popular sites (in fact, we have ordered them by popularity and omitted the odd ones leaving the even ones). For all purposes we treat them as unknown sites except that we now can check if their DNS-related IP addresses are assigned to other sites. As we see, the system still works well. Some IP addresses from the control group sites are assigned wrongly but the IP precision remains over 90% for both methods with the selected thresholds. All this suggests that the non DNS-related assigned IP addresses are correct in most cases.

VII. Conclusions

In this paper we have presented a method to label server IP addresses related to a predefined list of websites in a traffic trace. This is a far from trivial problem as users often access more than one website at the same time. Our initial motivation was to obtain labelled traffic traces that could be used to tune and test a web traffic classification system. Nevertheless, from the point of view of a network administrator, our system can also be used to monitor the traffic generated by specific websites or to identify the traffic directed to certain server IP addresses.

Our system labels individual IP addresses based on the number of times connections to them appear in sessions of a particular website against the total number of apparitions in the trace. It also labels IP subnetworks if various IP addresses that belong to them appear repeatedly in sessions of a website. We have tested our system with two traffic traces of, at least, a duration of ten days. We have identified an average of more than 20 IP addresses per website in each of the traces. We have validated our system by considering the assignation results of IP addresses that are related to the websites by DNS information obtaining good results.

References

[1] J. Charzinski, "Traffic properties, client side cachability and CDN usage of popular web sites," in *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 5987, pp. 136–150.

[2] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the world-wide web," *Computer Networks and ISDN Systems*, vol. 27, pp. 1065–1073, April 1995.

[3] E. B. Claise, "RFC 5101: Specification of the IPFIX protocol for the exchange of IP traffic flow information," Jan. 2008.

[4] A. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2005, vol. 3431, pp. 41–54.

[5] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: automated construction of application signatures," in *2005 ACM SIGCOMM workshop on Mining network data*. NY, USA: ACM, 2005, pp. 197–202.

[6] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 4, pp. 56–76, 2008.

[7] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network: measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, 2009.

[8] B. Yu and H. Fei, "Multiscale analysis and modeling of user session traffic in social networks," in *Proceedings of the 11th IEEE International Conference on Communication Technology*, nov. 2008, pp. 85 –88.

[9] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger, "Understanding online social network usage from a network perspective," in *Proceedings of the 9th Conference on Internet Measurement (IMC '09)*. New York, NY, USA: ACM, 2009, pp. 35–48.

[10] F. Schneider, S. Agarwal, T. Alpcan, and A. Feldmann, "The new web: Characterizing AJAX traffic," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 4979, pp. 31–40.

[11] D. Schatzmann, W. Mühlbauer, T. Spyropoulos, and X. Dimitropoulos, "Digging into HTTPS: flow-based classification of webmail traffic," in *Proceedings of the 10th Conference on Internet Measurement (IMC '10)*. New York, NY, USA: ACM, 2010, pp. 322–327.

[12] R. Archibald, Y. Liu, C. Corbett, and D. Ghosal, "Disambiguating HTTP: Classifying web applications," in *7th Wireless Communications and Mobile Computing Conference*, july 2011, pp. 1808 –1813.

[13] I. N. Bermudez, M. Mellia, M. M. Munafo, R. Keralapura, and A. Nucci, "DNS to the rescue: discerning content and services in a tangled web," in *Proceedings of the 2012 ACM conference on Internet Measurement*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 413–426.

[14] L. Torres, E. Magana, M. Izal, and D. Morato, "Identifying sessions to websites as an aggregation of related flows," in *Telecommunications Network Strategy and Planning Symposium*, 2012, pp. 1–6.

[15] ——, "Strategies for automatic labelling of web traffic traces," in *IEEE 37th Conference on Local Computer Networks*, 2012, pp. 196–199.

[16] "Argus: Audit Record Generation and Usage System," http://www.qosient.com/argus/.