Mini-IPC: A Minimalist Approach for HTTP Traffic Classification using IP Addresses

Pedro Casas and Pierdomenico Fiadino
Telecommunications Research Center Vienna - FTW
{surname}@ftw.at

Abstract—**The popularity of web-based services and multimedia applications like YouTube, Google Web Search, Facebook, and a bewildering range of Internet applications has taken HTTP back to the pole position on end-user traffic consumption. Today's Internet users exchange most of their content via HTTP. In this paper we address the problem of on-line HTTP traffic classification from network measurements. Building on the results provided by HTTPTag, a flexible system for on-line HTTP classification, we present and explore Mini-IPC. Mini-IPC is a minimalist approach for classifying HTTP flows using only the IP addresses of the servers hosting the corresponding content. Using one full week of HTTP traffic traces collected at the mobile broadband network of a major European ISP, we investigate to which extent the most popular HTTP-based services are hosted by well-defined sets of IP addresses, and evaluate the performance of Mini-IPC to classify these services using IPs only.**

Keywords*—*HTTP Traffic; Traffic Classification and Analysis; IP Addressing Space; CDNs; Mobile Networks' Traffic.

I. Introduction

HTTP is doubtlessly the dominating content delivery protocol in today's Internet. The popularity of services running on top of HTTP makes that more than 75% of today's residential customers traffic is accountable to HTTP [1], [2]. A big share of today's Internet ecosystem is shaped by the success and influence of the most popular web services: YouTube, Netflix, Facebook, and even Dropbox are forcing the Internet to shift the content as close as possible to the end-users, which in turn is modifying the way content is hosted, replicated, addressed, and served. The very last few years have seen an astonishing development in Content Delivery Networks (CDNs) technology, and it's not surprising that todays' Internet content is largely served by major CDNs like Akamai or Google CDN. In this scenario, understanding HTTP traffic composition, usage patterns, and content location and distribution is highly valuable for network operators.

In this paper we present Mini-IPC, a minimalist approach for classifying HTTP flows relying exclusively on IP addresses. Mini-IPC uses a single flow feature to classify HTTP flows: the IP address of the server hosting the corresponding content. Such an approach is extremely light-weight and can be applied for on-line high speed classification of HTTP traffic. Commonly deployed traffic classification methods rely on port and payload-based analysis techniques, both well-known in the field of network traffic classification. These techniques present important limitations that highly reduce their effectiveness, particularly due to the emergence of new dynamic applications and the widespread use of encryption, tunneling, and protocol obfuscation. The question is therefore why to use only the source IP address for classifying HTTP traffic, or more specifically, to which extent such an approach can provide useful results? The reason behind using IP addresses is simple: Mini-IPC targets only HTTP traffic, and services running on top of HTTP are provided by companies whose delivery infrastructure (i.e., servers hosting the content) tend to be either very stable in time, or in the case of CDN-based distribution, use well-known IP ranges.

Mini-IPC uses as *learning* input data the results provided by HTTPTag [12]. HTTPTag is a flexible on-line HTTP classification system based on pattern matching and tagging, which associates a set of labels or *tags* to each observed HTTP flow, based on the contents and service being requested. This association is performed by simple regular expressions matching, applied to the `host` field of the corresponding HTTP flow's header (i.e., host name of the contacted server). HTTPTag currently recognizes and tracks the evolution of more than 280 services and applications running on top of HTTP, including for example tags such as `YouTube`, `Facebook`, `Google` (i.e., Google Search), `Twitter`, `Zynga`, `Gmail`, etc. Due to the highly concentrated traffic volume on a small number of heavy hitter applications, current list of services spans more than 70% of the total HTTP traffic volume on the 3G network of a leading European provider. Figure 1 shows the deployment of HTTPTag and the Mini-IPC analysis algorithms in this network, using the METAWIN passive monitoring system [15] for traffic capturing, filtering, and analysis.

The goal of this paper is to evaluate the feasibility of using a simple approach as the one used by Mini-IPC to classify HTTP traffic on the fly. To this end, we study the dynamics of content hosting and services addressing in current Internet, and evaluate the performance of Mini-IPC. Using a week of HTTP traffic traces collected at the mobile broadband network of a major European ISP, we study the associations between services, the hosting organizations, and the IPs assigned to the servers providing the content. The complete dataset consists of passively observed HTTP flows, aggregated in a per-hour basis. For each query, the dataset contains the contacted URL, the contacted IP address, and a timestamp. The Full Qualified Domain Name (FQDN) is automatically extracted from the URL, which is then used to deduce the corresponding service being accessed at the contacted IP, using HTTPTag. We additionally gather the total volume exchanged with the contacted IP, and using the MaxMind ASes databases (http://www.maxmind.com) we include the name of the organization owning the contacted IP.

978-1-4673-2480-9/13/$31.00 ©2013 IEEE

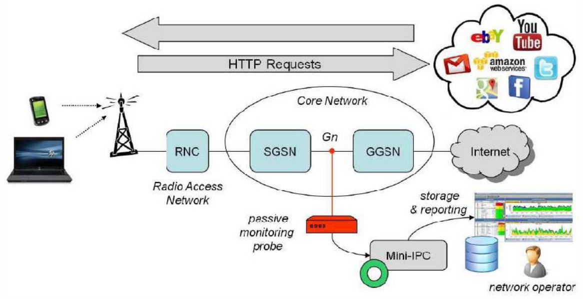


Figure 1. Deployment of Mini-IPC in an operational 3G Network.

The remainder of the paper is organized as follows: section II presents a brief state of the art in the field of automatic traffic analysis and classification, specially focusing on HTTP traffic. In section III we describe the labeling technique used by HTTPTag to generate the ground-truth for the Mini-IPC study. Section IV presents Mini-IPC and provides a comprehensive analysis of the top services consumed in the network under study, which form the classes used by Mini-IPC in this work. In section V we present the classification performance of Mini-IPC in the labeled HTTP flows, including its global accuracy and the per-service recall and precision for the top services in the traces. Finally, section V concludes this work.

II. Related Work

The field of automatic Internet Traffic Classification (TC) and analysis has been extensively studied during the last decade [5]. Standard classification approaches rely on Deep Packet Inspection (DPI) techniques, using pattern matching and statistical traffic analysis [6]. Probably the most popular approach for TC exploited in recent years by the research community is the application of Machine Learning (ML) techniques [7]–[10].

In the specific case of HTTP traffic, classification and analysis has been the focus of many recent studies [2]–[4], [12], [13]. In [12] we use pattern matching techniques applied to the `host` field of HTTP headers to recognize more than 280 applications and services running on top of HTTP. In [2], [4], authors use DPI techniques to analyze the usage of HTTP-based applications on residential connections, showing that HTTP traffic highly dominates the total downstream traffic volume. Authors in [3] study the extension of HTTP content caching in current Internet, characterizing HTTP traffic in 16 different classes using port numbers and heuristics on application headers. Recently, the authors of [13] provide evidence on a number of important pitfalls of standard HTTP traffic characterization techniques which rely exclusively on HTTP headers, showing for example that around 35% of the total HTTP volume presents a mismatch in headers like `Content-Type`, extensively used in previous studies.

In this paper we present Mini-IPC through the analysis of a week of HTTP traffic traces collected at the 3G network of a major European ISP. The main purpose of this study is not to provide a highly accurate classifier for HTTP traffic flows at the Internet-wide scale. Rather, the goal is to explore the possibility of using only IP addresses for classifying HTTP traffic flows, offering a practical and very flexible solution for traffic classification and traffic aware networking. We acknowledge that, despite the large size of our traffic dataset, we analyze packets from a single vantage point, which is far from providing a complete view of the global IP address space and HTTP services.

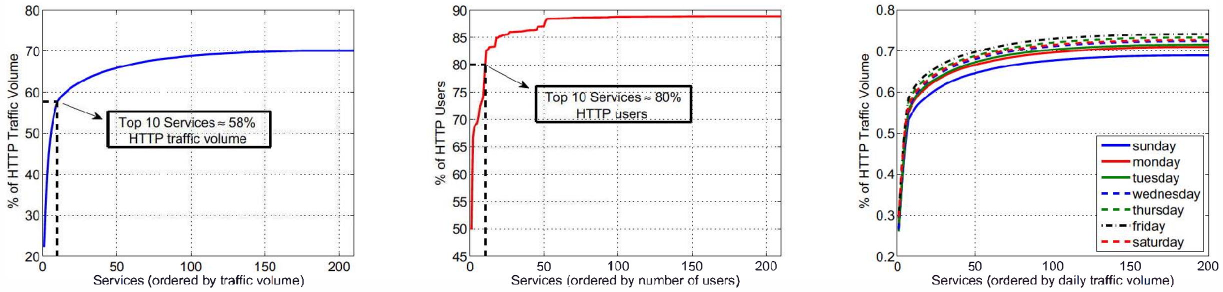
III. HTTPTag: Linking IPs and Services

In this section we provide a description of the HTTPTag system, which is used to label the traffic dataset presented on this paper. HTTPTag works with packet data, passively captured at the vantage point of analysis. Packets are captured on the Gn interface links between the GGSN and SGSN nodes. HTTP packets are detected and analyzed on the fly: every new HTTP transaction is parsed and the contacted host name is compared against a set of defined regular expressions or *patterns* describing different services and applications. If a matching pattern is found, the transaction is assigned to the corresponding service. To preserve user privacy, any user related data (e.g., IMSI, MSISDN) are removed on-the-fly, and payload content beyond HTTP headers is ignored.

HTTPTag uses TicketDB [14], a fast and scalable parallel database system tailored to meet the requirements of network monitoring in 3G networks. For every new HTTP transaction analyzed by HTTPTag, a summary ticket is stored and indexed in TicketDB, providing long term traffic analysis capabilities to the system. Each ticket contains a timestamp, the IP address of the contacted server, the requested URL, volume stats (i.e., transferred bytes up/down), and the corresponding service resulting from the pattern matching step. As such, for every observed HTTP flow, HTTPTag provides a mapping between the hosting IP address and the corresponding service.

To improve pattern matching, patterns are ordered by probability of occurrence, which are computed from the history of successful matches. HTTPTag tagging approach is based on manual definition of tags and regular expressions, which might a priori impose scalability issues. Indeed, there are millions of websites on the Internet and it would be impossible to define enough patterns to classify every possible requested URL. However, the well known mice and elephants phenomenon also applies to HTTP-based services, and limiting the study to the most popular services already captures the majority of the traffic volume/users in the network. While the initial definition of tags is a time-consuming task, regular expressions identifying applications tend to remain stable in time, basically because they are associated to the name of the application itself and thus recognized and used by the end-user. This is specially true for popular services, which carry the most of the traffic. HTTPTag does not currently recognize HTTPS traffic, since the requested URLs are encrypted. An on-going extension of HTTPTag to solve this issue is to rely on DNS queries analysis, similar to the approach introduced in [11].

Figures 2(a) and 2(b) depict the distribution of HTTP traffic volume and number of users covered by HTTPTag in a standard day (outside the week of analysis). Using about 380 regular expressions and 280 tags (i.e. services) manually defined, HTTPTag can classify more than 70% of the overall HTTP traffic volume caused by more than 88% of the web users in the studied network. As previously mentioned, a small number of heavy hitter services dominates the HTTP landscape: the top-10 services w.r.t. volume account for almost



(a) HTTP traffic volume per service.

(b) Unique HTTP users per service.

(c) Daily HTTP traffic volume per service.

Figure 2. HTTPTag labels more than 70% of the overall HTTP traffic caused by more than 88% of the web users. The top-10 services w.r.t. volume account for almost 60% of the overall HTTP traffic, and the top-10 services w.r.t. popularity are accessed by about 80% of the users.

60% of the overall HTTP traffic, and the top-10 services w.r.t. popularity are accessed by about 80% of the users. These results reinforce the hypotheses behind HTTPTag: focusing on a small portion of the services already gives a large traffic visibility to the network operator. Figure 2(c) shows the total daily HTTP volume labeled by HTTPTag on the week of traces used in the study. The week corresponds to the first 7 days of April 2012, from Sunday the 1st till Saturday the 7th. HTTPTag is able to label between 69% and 74% of the total daily HTTP volume on the studied traces. In the following sections we only consider labeled flows, and the approximately remaining 30% of unlabeled HTTP volume is discarded.

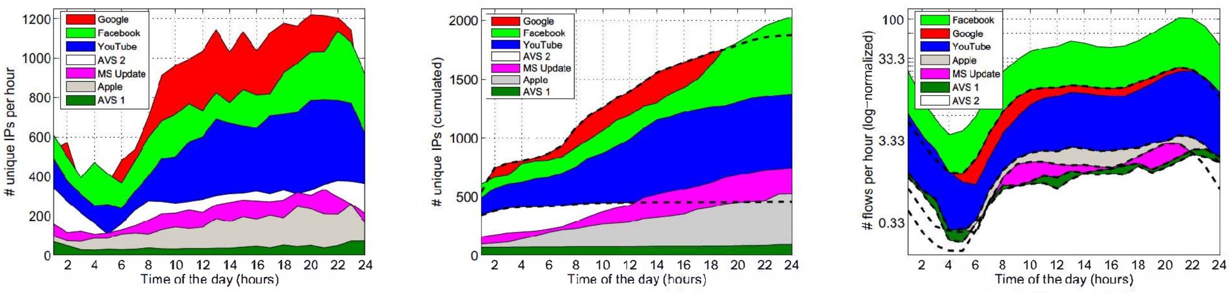
IV. MINI-IPC AND TRAFFIC ANALYSIS

Mini-IPC classifies HTTP flows based solely on the IP address of the server being contacted. In a nutshell, given a specific service S_i to be identified, Mini-IPC builds a set of k_i well-known IP addresses $IP_i = \{ip_i(1), ip_i(2), \ldots, ip_i(k_i)\}$ hosting S_i, using the associations $A_i = \{S_i.IP_i\}$ between server IPs and services provided by HTTPTag on a certain *learning* period. Given a list of m services $S_{i,\{i=1..m\}}$ to classify and a downstream HTTP flow f_{new} coming from IP address ip_{new}, Mini-IPC applies the following classification rule: $\mathcal{F}(f_{new}) = S_i \leftrightarrow ip_{new} \in IP_i$. Given the widespread usage of third-party hosting organizations serving the content of multiple services (e.g., Akamai), the big number of companies hosting multiple services in the same datacenters (e.g., Google), and the ISPs content caching policies, multiple different services S_i might be associated to the same IP address. Therefore, the m sets IP_i are not necessarily disjoint sets. We shall refer to this IP sets intersection issue as IP *hosting collisions*. In this case, the previous classification rule would associate f_{new} to all those services mapped to ip_{new}. To solve this multi-classification issue and decide for one single output, Mini-IPC currently uses a random selection approach, in which the decided service is randomly chosen among the potential ones. Such a straightforward decision approach could be improved by heuristics, for example by adding weights to the candidate services based on the size of the IP sets.

Before going into the evaluation details of Mini-IPC and to better understand the ideas behind its classification rule, we devote the remaining part of this section to provide the results of a comprehensive analysis we have performed on the dataset under study. Let us first focus the attention on the aforementioned top-10 services w.r.t. volume depicted in figure 2(a). These top-10 services are responsible for almost 60% of the total daily HTTP volume during the whole evaluation week, which represents about 85% of the labeled services in terms of traffic volume. The ordered list of services in terms of volume includes YouTube (YT), Facebook (FB), Google (i.e., Google Search - GO), Apple (i.e., App Store and iTunes - APP), two well-known Adult Video Streaming services AVS 1 and AVS 2, Microsoft Windows Update - WIN, and three more video streaming services. The first 7 services will form the basis of our study.

Let us now focus on the number of unique IPs addresses used by each of these top-7 services on a single day. Figures 3(a) and 3(b) depict the evolution of the number of unique IPs per hour and the accumulated number of unique IPs on a single day, whereas figure 3(c) plots the number of HTTP flows per hour (values are normalized for privacy reasons). For 6 out of the 7 services, there is a clear correlation between usage and number of unique IPs providing the corresponding servers. The changes observed in the unique number of IPs being used by Google Search, Facebook, and YouTube is impressive, going from about 250 IPs per service at 5 am to up to 1200 in the case of Google Search. These three servers are provided by large CDNs (i.e., Google CDN for Google services and Akamai for Facebook), which justifies the large number of unique IPs being used during the day. Thanks to Akamai, Facebook is the most IP-distributed service, using more than 2000 different IPs on a single day. The number of unique IPs serving the video streaming service AVS 1 remains almost constant in time and is below 100 all over the day, suggesting a very stable delivery infrastructure. Using the MaxMind ASes databases we explore now how distributed are these unique IPs in terms of the different organizations owning them. Figure 4(a) shows the fraction of unique IPs per service hosted by the list of organizations and ASes described in table I. The organization labeled as "other" (i.e., id o) consists mainly of ISP ASes which cache the content on the edge of their own networks.

As expected, Google Search and YouTube IPs are mainly hosted by Google Inc. ASes, Facebook IPs are mainly hosted by Akamai and Facebook ASes, and Windows Update IPs are mainly hosted by Microsoft ASes. For example, in the case of Facebook, it is well known that the static content is hosted by Akamai, whereas Facebook ASes host the dynamic content.

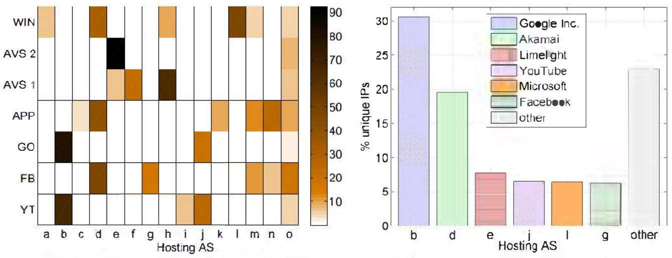


(a) Unique IPs per hour. (b) Cumulative number of unique IPs. (c) Number of flows per hour.

Figure 3. Evolution of unique IPs and num. of flows for the top-7 services on a single day. Google Search, Facebook and YouTube dominate the IP space and account for the majority of the flows. Thanks to Akamai, Facebook is the most IP-distributed service, using more than 2000 different IPs on a single day.

Org. (AS num.)	id	Org. (AS num.)	id	Org. (AS num.)	id
Hotmail (12076)	a	Swiftwill (30361)	f	Apple (714)	k
Google (15169)	b	Facebook (32934)	g	Microsoft (8075)	l
Omniture (15224)	c	Level 3 (3356)	h	TeliaNet (1299)	m
Akamai (20940)	d	YouTube (36040)	i	Verizon (701)	n
Limelight (22822)	e	YouTube (43515)	j	other	o

Table I. TOP HOSTING ORGANIZATIONS AND ASES IN TERMS OF NUMBER OF UNIQUE IPS OF THE T●P-10 SERVICES (NON-ORDERED LIST).



(a) Services & hosting organizations. (b) Share of IPs per organizations.

Figure 4. Distribution of the server IPs used by the top 7 services among the top hosting organizations.

Almost all of the AVS 2 IPs are hosted by Limelight, and this organization is additionally hosting only a small fraction of AVS 1 IPs, with no other service being hosted there. This concentration of IPs on an almost exclusive organization explains the high classification accuracy obtained by Mini-IPC for AVS 2 flows in section V. In all the cases, a small fraction of the IPs used to deliver the services belong to ASes caching the content. The most interesting observation is that many IPs of different services are usually hosted by the same organization, resulting on potential IP hosting collisions. For example, Akamai hosts content from Facebook, Apple, and Windows Update, whereas both YouTube and Google Search belong to Google and YouTube ASes. Figure 4(b) shows that such potential IP hosting collisions are actually pretty high, as most of the unique IPs are hosted by Google Inc. and Akamai. Google and Akamai are clearly the most distributed organizations in terms of IPs providing the top HTTP services. Limelight is the third CDN in our traces, in this case mainly providing the AVS 2 content.

To further explore this IP hosting collisions issue, figure 5 depicts the distribution of the IP address ranges associated to the different top-7 services on a single day. The plots additionally include the distribution of the unique IPs used by all the rest of the labeled services. Figure 5(a) depicts the complete range of IPs, whereas figures 5(b) and 5(c) zoom on specific ranges showing IP hosting collisions. The number of collisions is pretty high: for example, figure 5(b) shows that about 8% of the Facebook IPs are in the same range of about 17% of Windows Update IPs and 3% of Apple IPs, and that about 16% of the IPs used by AVS 1 also intersect with 1% of Windows Update IPs. Figure 5(c) additionally shows IP hosting collisions between Google Search and Facebook, AVS 1 and AVS 2, and among Facebook, Apple, and Windows Update on a different IP range. These collisions are the main reason of the low Recall and Precision achieved by Mini-IPC on many of these services.

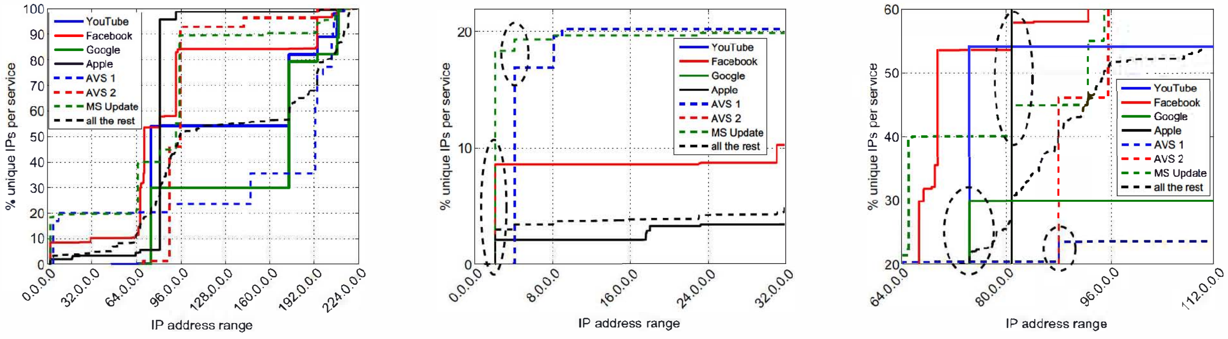
V. USING MINI-IPC FOR TRAFFIC CLASSIFICATION

The last part of the study is devoted to the evaluation of Mini-IPC as a traffic classifier. In order to test the classification performance achieved for each of the previously analyzed top-7 services, we divide the complete week of labeled HTTP flows in $n = 8$ services or *classes*: the first 7 correspond to the top-7 services, whereas the 8th class corresponds to all the rest of the labeled flows and will be referred to as the `other` class. Using the labeled traffic flows of Monday as learning dataset, we construct 7 IP sets IP_i containing all the unique IPs per service observed during the day. The size of each of this sets is available in figure 3(b): $\{\#IP_i\} = \{1373, 2\bullet 31, 1875, 522, 92, 456, 743\}$. The classification associated to the class `other` is simply done by a complementary decision rule: if according to $\mathcal{F}(f_{new})$, flow f_{new} is not assigned to any of the top-7 services, then it is assigned to the `other` class.

To evaluate the classification performance of Mini-IPC, we employ three traditionally used performance metrics in the traffic classification literature: the Classification Accuracy (CA), and the Recall (R_i) and Precision (P_i) per class:

$$\text{CA} = \frac{\sum_{i=1}^{m} TP_i}{n}, \quad \text{R}_i = \frac{TP_i}{TP_i + FN_i}, \quad \text{P}_i = \frac{TP_i}{TP_i + FP_i}$$

where TP_i corresponds to the number of correctly classified flows in class i (i.e., number of true positives), and FN_i and FP_i correspond to the number of false negatives and false positives in class i. The classification accuracy indicates the

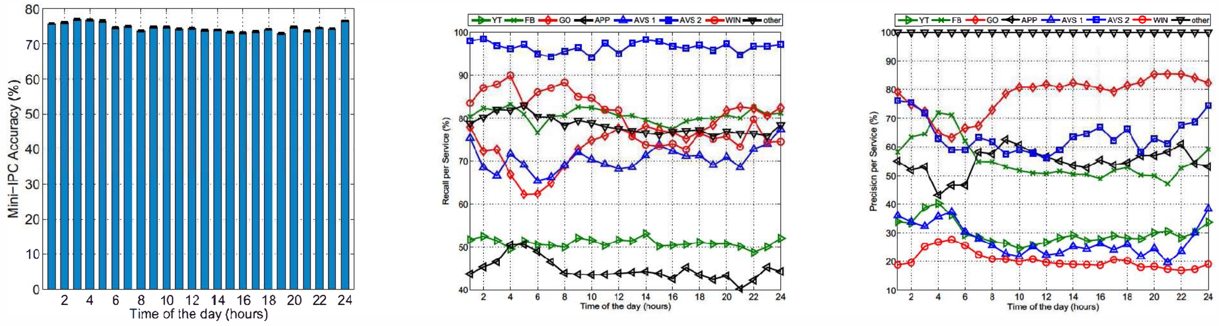


(a) IP range distribution on a single day.

(b) Service/IP hosting collisions.

(c) Service/IP hosting collisions.

Figure 5. Distribution of the IP range associated to the tagged services on a single day. Different services are hosted by the same organization.



(a) Mini-IPC Accuracy.

(b) Mini-IPC Recall per Service.

(c) Mini-IPC Precision per Service.

Figure 6. Classification performance of Mini-IPC in the learning day. The classification accuracy is high and stable during the day, close to 75% of correctly classified HTTP flows. More than 60% of all the Facebook, Adult Video, Google Search, and Windows Update HTTP flows are correctly classified.

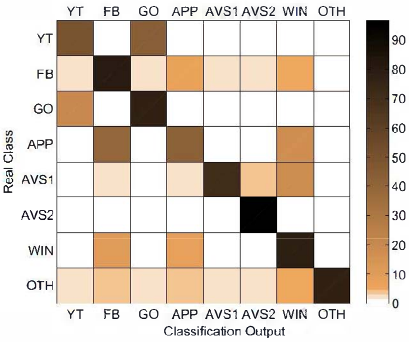
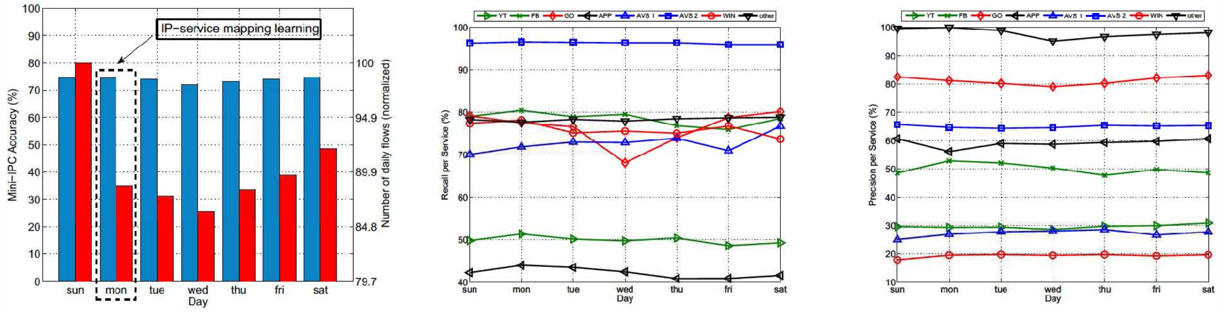


Figure 7. Confusion matrix for Mini-IPC traffic classification.

percentage of correctly classified flows among the total number of flows n. The recall R_i measures the per-class accuracy, whereas the precision P_i measures how precise is the classifier in labeling flows from a given class only.

Figure 6 depicts the classification performance achieved by Mini-IPC in the learning day (i.e., Monday), on an hourly basis. Given the random decision process used by Mini-IPC in case of IP hosting collisions, the algorithm is run 20 consecutive times, and the provided results correspond to the obtained average values. Figure 6(a) depicts the classification accuracy for the 8 defined classes, including the error variance bounds resulting from the 20 consecutive runs, which are negligible. The overall classification accuracy is remarkably high and stable during the day, rounding about 75% of correctly classified HTTP flows. These a-priori excellent results achieved by only using IP addresses can be in fact misleading, because we are considering the `other` class inside the classification process, which contains a much higher number of unique IPs as revealed by figure 5(a). Figure 7 shows the confusion matrix for the classification results. Many YouTube flows are classified as Google Search, and vice versa. Windows Update flows are misclassified as Facebook and Apple, given the previously mentioned IP hosting collisions within Akamai. Similar behavior is observed for Apple. As previously observed, the AVS 2 service is accurately classified with a very low false negatives rate.

Let's focus now on the per service recall and precision, depicted in figures 6(b) and 6(c) respectively. The recall or per-service classification accuracy is still remarkably high and stable during the day, with more than 60% of all the Facebook, Adult Video Streaming, Google Search, and Windows Update HTTP flows correctly classified. Specially in the case of the AVS 2 service, recall is as high as 98%, and both Facebook and Windows Update HTTP flows are identified with a per-class accuracy above 80%. YouTube and Apple flows are poorly classified, and the recall achieved is between 40% and 50%. The main reason for these poor results come directly



(a) Mini-IPC Daily Accuracy and num Flows.

(b) Mini-IPC Daily Recall per Service.

(c) Mini-IPC Daily Precision per Service.

Figure 8. Classification performance achieved by Mini-IPC in the analyzed week of IITTP traffic. Mini-IPC accuracy is stable during the complete week.

from the IP hosting collisions associated to Google CDN and Akamai, as many of the YouTube and Apple flows are classified as Google Search and Facebook or Windows Update flows respectively, as depicted in figure 7.

When it comes to evaluate the per-service precision, the achieved results are much less encouraging. The recall obtained for Google flows is still pretty high and above 80% from 9 am onwards, but results for YouTube, AVS 1, and Windows Update show a big number of false positives associated to these services. As expected, the precision for the `other` class is of 100% during the complete learning day, which comes directly from the applied classification technique for this specific class.

Thc final analysis stcp consists in thc classification pcrformance of Mini-IPC on the complete week of flows, using the IPs of Monday as learning data. Figure 8 depicts the per-day accuracy, recall and precision achieved in the 7 days of the study. Figure 8(a) shows that the classification accuracy is remarkably stable during the full week, clearly suggesting that the sets of IPs serving by the different services are stable in time, at least in a weekly-basis. The figure additionally shows the normalized number of analyzed flows per day, to have an idea of the volume variations during the week. Figures 8(b) and 8(c) additionally present the daily recall and precision for the full week, showing once again that classification performance is very stable in time. In fact, achieved results remain almost unchanged from those obtained during the training day, achieving a classification accuracy close to 75%.

VI. Concluding Remarks

In this paper we have addressed the problem of HTTP traffic classification from network measurements, exploring Mini-IPC, a minimalist approach for classifying HTTP flows using only the IP addresses of the servers hosting the corresponding content. Using one full week of HTTP traffic traces collected at the mobile broadband network of a major European ISP, we have investigated the associations between services, the hosting organizations, and the IPs assigned to the servers providing the content, evaluating the performance of Mini-IPC to classify the top services accessed by the users of the studied network. Among our main findings, we have shown that despite its simplicity, Mini-IPC was able to classify the HTTP flows of the top services with a classification accuracy as high as 75%. However, we have also seen that the classification recall and precision are highly impacted by IP hosting collisions, seriously impacting the performance of Mini-IPC as a robust traffic classifier. Still, results obtained for some of the analyzed services like Google Search, Facebook, Windows Update, and AVS services were encouraging, achieving a daily per-class accuracy above 70% in all the cases, with precision values above 65% for Google Search and AVS 2. This paper has therefore provided evidence on the possibilities of using Mini-IPC for recognizing the top HTTP services in terms of end-user consumed traffic volumes, offering a practical and very flexible solution for traffic aware networking.

Acknowledgments

This work has been performed in the framework of the EU-IP project mPlane, funded by the European Commission under the grant 318627. Part of the work has been done in the framework of the projects Darwin3 and N-0 at the Telecommunications Research Center Vienna (FTW), and has been partially funded by the Austrian Government and the City of Vienna through the program COMET.

References

[1] A. Gerber and R. Doverspike, "Traffic Types and Growth in Backbone Networks", in *OFC/NFOEC*, 2011.

[2] G. Maier, A. Feldmann, V. Paxson, ans M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic", in *ACM IMC*, 2009.

[3] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, and O. Spatscheck, "Network-Aware Forward Caching", in *WWW*, 2009.

[4] J. Erman, A. Gerber, and S. Sen, "HTTP in the Home: It is not just about PCs", in *ACM CCR* 41(1), 2011.

[5] A. Dainotti, A. Pescapé, and K. C. Claffy, "Issues and Future Directions in Traffic Classification", in *IEEE Network*, 2012.

[6] A. Finamore et al., "Experiences of Internet Traffic Monitoring with Tstat", in *IEEE Network* 25(3), 2011.

[7] T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning", in *IEEE Comm, Surv. & Tut.*, 2008.

[8] A. Moorc and D. Zucv, "Intcrnct Traffic Classification using Baycsian Analysis Techniques", in *ACM SIGMETRICS*, 2005.

[9] N. Williams cl al., "A Prcliminary Pcrformancc Comparison of Fivc Machinc Learning Algorithms for Practical IP Traffic Flow Classification", in *ACM CCR*, vol. 36 (5), 2006.

[10] P. Casas, J. Mazel, P. Owezarski, "MINETRAC: Mining Flows for Unsupervised Analysis & Semi-Supervised Classification", in *ITC*, 2011.

[11] I. Bermudez et al., "DNS to the rescue: Discerning Content and Services in a Tangled Web", in *ACM IMC*, 2012.

[12] P. Fiadino, A. Bär, P. Casas, "HTTPTag: A Flexible On-line HTTP Classification System for Operational 3G Networks", in *IEEE INFOCOM*, 2013

[13] F. Schneider et al., "Pitfalls in HTTP Traffic Measurements and Analysis", in *PAM*, 2012.

[14] A. Bär, A. Barbuzzi, P. Michiardi, F. Ricciato, "Two Parallel Approaches to Network Data Analysis", in *LADIS*, 2011.

[15] F. Ricciato, "Traffic Monitoring and Analysis for the Optimization of a 3G Network", in *IEEE Wireless Communications*, vol. 13(6), 2006.