Distilling the Internet's Application Mix from Packet-Sampled Traffic

Philipp Richter[1], Nikolaos Chatzis[1], Georgios Smaragdakis[2,1], Anja Feldmann[1], and Walter Willinger[3]

[1] TU Berlin [2] MIT [3] NIKSUN, Inc.

Abstract. As the Internet continues to grow both in size and in terms of the volume of traffic it carries, more and more networks in the different parts of the world are relying on an increasing number of distinct ways to exchange traffic with one another. As a result, simple questions such as "What is the application mix in today's Internet?" may produce non-informative simple answers unless they are refined by specifying the vantage point where the traffic is observed, the networks that are involved, or even the type of interconnection used.
In this paper, we revisit the question of the application mix in today's Internet and make two main contributions. First, we develop a methodology for classifying the application mix in packet-sampled traces collected at one of the largest IXPs in Europe and worldwide. We show that our method can classify close to 95% of the traffic by relying on a *stateful* classification approach that uses payload signatures, communication patterns, and port-based classification only as a fallback. Second, our results show that when viewed from this vantage point and aggregated over all the IXP's public peering links, the Internet's application mix is very similar to that reported in other recent studies that relied on different vantage points, peering links or classification methods. However, the observed aggregate application mix is by no means representative of the application mix seen on individual peering links. In fact, we show that the business type of the ASes that are responsible for much of the IXP's total traffic has a strong influence on the application mix of their overall traffic and of the traffic seen on their major peering links.

1 Introduction

Knowing the Internet's application mix is important for tasks such as identifying the emergence of new trends in Internet usage, optimizing application performance, and provisioning network resources. As a result, there exists a growing body of literature on inferring the Internet's application mix, with the different papers typically relying on different data sources and deploying different traffic classification techniques (e.g., see [13, 25, 29] and references therein).

However, due to the heterogeneity of the Internet and its complex topology and global scope, there are no simple answers to questions like "What are the most popular applications in today's Internet?" or "What is the application mix in today's Internet?" In fact, as more and more networks consider factors such as cost, performance, security, ease-of-use, and flexibility when deciding about which kind of traffic to send over which type of peering links, the application mix can be expected to differ from link to link.

In this paper, we are primarily interested in how representative commonly-reported aggregate statistics concerning the Internet's application mix are in view of the network's enormous heterogeneity. To this end, we first develop a new methodology to classify traffic from packet-sampled traffic traces. Packet sampling is a widely employed technique when monitoring high-bandwidth infrastructures and is commonly used by large ISPs and IXPs. We then rely on traffic traces collected at such a large IXP and apply our traffic classification methodology to infer the application mix on tens of thousands of public peering links at this IXP.[1] Our results show that the heterogeneity of the Internet extends directly to the application mix of its traffic, and we illustrate the observed heterogeneity by providing insight into how and why the application mix can differ from interconnection to interconnection and among different types of networks.

Our contributions can be summarized as follows:

- We develop a traffic characterization methodology that is able to classify up to 95% of the traffic in our dataset (i.e., peering traffic exchanged at a large IXP, see Section 2). The novelty of our methodology is that it uses a *stateful* classification technique (i.e., it keeps track of classified connection endpoints) that is by and large able to overcome the challenges posed by random packet sampling (see Section 3).
- We apply our new methodology to a set of traffic traces collected at a large European IXP over a period of 2.5 years and provide details about the aggregate application mix seen at this IXP, including pronounced diurnal cycles as well as trends that become visible when monitoring the application mix over time (see Section 4).
- We compare the aggregate application mix observed at our IXP to that reported in other recent studies, which use different techniques and vantage points. We find that when aggregated over all of the IXP's peering links, the observed application mix is comparable to the application mix reported in these studies. However, we also show that the aggregate application mix is by no means representative of the application mix seen on an individual peering link and that the business type of the networks on either side of these peering links has a strong influence on the application mix of the traffic that traverses those links (see Section 5).

2 Dataset Characteristics

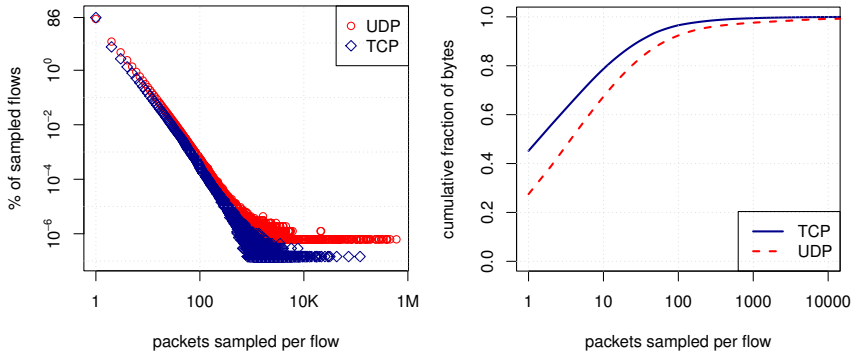
In this paper, we rely on packet-sampled traffic traces captured from the public switching fabric of a large European IXP. We use five snapshots (selected from a period that spans 2.5 years), each covering a full week (168 consecutive hours). Table 1 lists the pertinent properties of these traces. Unless mentioned otherwise, we use the most recent snapshot (i.e., 09-2013) as default dataset.

During the most recently monitored period in September 2013, the IXP had close to 500 members and a peak traffic rate close to 2.5 Tbps. Our traces consist of sFlow [28] records, captured using a random packet sampling rate of 1-out-of-16K (2^{14}) packets. For more details on the sampling process and the IXP's peering link characteristics, see [7, 27]. sFlow captures the first 128 bytes for each Ethernet frame. Thus, each packet includes the full link layer (Ethernet), network layer (IP), and transport layer

[1] Traffic traversing the IXP's private peering links is not collected and not considered here.

Name	Timerange	Sampling	Packets	Bytes	IPv4 / IPv6	TCP / UDP
09-2013	2013-09-02 to 2013-09-08	1/16K	9.3B	5.9TB	99.36 / 0.63	83.7 / 16.3
12-2012	2012-12-01 to 2012-12-07	1/16K	8.5B	5.5TB	99.64 / 0.36	83.1 / 16.9
06-2012	2012-06-04 to 2012-06-10	1/16K	7.3B	4.6TB	99.80 / 0.20	80.7 / 19.3
11-2011	2011-11-28 to 2011-12-04	1/16K	6.4B	4.2TB	99.93 / 0.07	79.8 / 20.2
04-2011	2011-04-25 to 2011-05-01	1/16K	5.3B	3.5TB	99.94 / 0.06	79.2 / 20.3

Table 1. Overview of dataset characteristics. The number of packets/bytes refer to the number of packets collected i.e., after sampling.



(a) Samples per flow (1200s timeout).

(b) Fraction of total bytes related to number of packets sampled per flow.

Fig. 1. IXP data sampling characteristics relevant for traffic classification.

(TCP/UDP) protocol headers, as well as a limited number of payload bytes. In the most common case, where the IPv4 and TCP protocols are used, this leaves 74 bytes worth of payload information (if TCP option fields are set, the available payload is further reduced by a few bytes). In the following, we consider only IPv4 traffic, as the fraction of IPv6 is still below 1% in all our snapshots.

The sampled nature of our datasets poses significant challenges when trying to apply traditional traffic-classification approaches (see Section 3 for details). To assess the impact of sampling on the visibility of "full" flows, we aggregate the packets sampled at our IXP using the typical 5-tuple aggregation consisting of source and destination IP addresses, source and destination port numbers, and the transport protocol. Figure 1(a) shows the number of packets that are sampled for each flow, using a 1200s timeout. It shows that we see only a single packet for some 86% of the sampled TCP flows (76% for sampled UDP flows). We also observe flows for which we sample several hundreds of thousands of packets over the course of one week. Surprisingly, UDP flows dominate the heavy-hitter flows and closer inspection reveals that most of the large UDP flows are related to recursive DNS interactions between name servers. Accordingly, Figure 1(b) shows the cumulative total number of bytes related to flows for which we sample less or equal than x packets. It shows that in case of TCP, more than 45% of the bytes are sampled from flows for which we sample only a single packet (27.5% for UDP). Since we only observe packets, we cannot rely on any per-flow properties nor can we expect to sample packets at any specific position of a flow e.g., the first packet(s). Moreover, we cannot expect to have any visibility into the bidirectional nature of any of the flows–all that sampling gives us is a "random set of packets."

3 Classification Approach

3.1 Related Work

Application classification has attracted the attention of researchers for many years and has resulted in a large number of different methods and studies. However, the characteristics of our datasets (i.e., sampling, no bidirectional visibility) pose new challenges for application classification. In particular, since most of the existing classification approaches require information that is not available in our datasets(e.g., unsampled packet traces, flow statistics), these methods are not directly applicable in our context.

Before presenting our new application characterization method, we first provide a condensed taxonomy of existing classification approaches. To this end, we follow closely the description presented in [20] and focus on those aspects of the different approaches that prevent them from being directly applicable to the types of datasets we are considering. For a more detailed discussion of the various existing application classification approaches, we refer to extensive surveys such as [10, 13, 20, 25, 29].

Port-based approach: Many applications typically run on fixed port numbers which can be leveraged to classify packets to their corresponding applications. The drawbacks of port-based classification are that *(i)* applications can rely on random port numbers (e.g., as Peer-to-Peer (P2P) applications) and *(ii)* applications might use well-known port numbers to obfuscate traffic (e.g., see [24]). On the positive side, port-based classification has been shown to be still effective [23], is robust to sampling and can be applied to our dataset in a straight-forward manner. Note that port-based classification was already performed for the sFlow data captured at this IXP in [7].

Payload-based approach: Also referred to as Deep Packet Inspection (DPI), payload-based classification produces very accurate results by relying on application-specific signatures (i.e., known byte patterns of known protocols). Application signatures are typically based on protocol handshakes and can often be assembled using only the first few payload-carrying packets that are exchanged between the communicating hosts (i.e., an `HTTP GET` request followed by an `HTTP/1.{0,1}` reply). The payload-based approach is often used to establish ground truth for the application mix of traffic traces (see e.g., [11] for a comparative study). While we have access to the initial bytes of the payload of each sampled packet, we do not necessarily sample the first packet(s) of flows that contain application signatures. In addition, we cannot inspect bidirectional payload patterns of flows using our datasets.

Flow features-based approach: By utilizing flow properties (e.g., the total number of packets, average packet size), several approaches focus on classifying flows as belonging to specific applications without inspecting the payload of packets. Since we do not have per-flow information, these approaches are not applicable to our datasets.

Host behavior-based approach: This class of approaches classifies traffic by profiling the detailed network interaction of hosts (e.g., which destinations are contacted on which ports [19] or the network-wide interactions of hosts [17]). The various approaches in this class have been shown the be particularly effective for characterizing P2P applications [18]. While we are not able to perform fine-grained profiling of hosts due to the sampled nature of our data, we do make use of properties inferred from the *social* behavior of hosts to uncover parts of Peer-to-Peer traffic.

3.2 Building Blocks

The foundation of our classification approach outlined below is the ability to attribute *some* of the sampled packets to their respective applications by mainly using payload signatures and partly relying on port numbers. In particular, we rely on signatures which we derived from the *L7-filter* [3] and the *libprotoident* library [8] for well-known protocols such as HTTP, SMTP, POP3, IMAP, NNTP and SSH. We also make use of application signatures derived from protocol specifications [1, 6] for BitTorrent. We also used available signatures to detect other P2P protocols (e.g., eDonkey) but their contributions in terms of classifying packets were insignificant. We verified all application signatures using manually generated traffic traces. For SSL-based protocols (we focus on HTTPS, NNTPS, POP3S, and IMAPS), we use signatures indicating an SSL handshake and consider SSL handshake packets on the well-known port number of the respective application (e.g., 443 for HTTPS) as belonging to that application.

To ensure the accuracy of our application signatures (i.e., keeping the false positives low by limiting the number of signatures), we restrict our set of application signatures and port numbers and only consider applications that *(i)* generate significant traffic and *(ii)* are reliably detectable using application signatures and, if needed, port numbers. For example, we do not try to classify Skype traffic because its detection remains unreliable unless specialized approaches are used [9].

3.3 Classification Method

Figure 2 illustrates our classification pipeline. In particular, our classification approach requires that the given traffic trace be processed twice, first in a *pre-classification* phase and then in a *classification* phase. The purpose of the first phase is to derive *state*, which will then be leveraged in the *classification* phase to attribute packets to their respective endpoints, revealing the corresponding application.

I. Pre-classification phase

The goal of the pre-classification step is to extract server *endpoints* and IP addresses of clients, which will be used as state in the subsequent classification phase. In this phase, we rely solely on payload-based classification using our validated signatures (as well as SSL signatures on well-known ports). For each packet that belongs to a client-server application, we save the server endpoint, i.e., its (IP, port) tuple. To identify the server-side of a packet, we rely on directed signatures (e.g., HTTP request vs. HTTP reply). For packets matching a BitTorrent signature, we save the SRC and DST IPs but not the port numbers. Since most BitTorrent traffic that matches our signatures is UDP-based which, due to its connectionless nature, is more susceptible to spoofing as well as other phenomena such as BitTorrent DHT poisoning for control traffic (e.g., [30]), we only count an IP address as BitTorrent speaker if we sample at least 2 packets that originate from/are sent to that IP address matching our signatures. Additionally, we save IP addresses of HTTP clients. In this pre-classification, we identify more than 2.7M HTTP server endpoints (1.43M unique IP addresses), and 210K HTTPS endpoints. On the client side, we identify 37.7M HTTP client IPs, as well as 38.9M BitTorrent speakers, where the overlap between HTTP client IPs and BitTorrent speakers is 12.4M IP addresses.

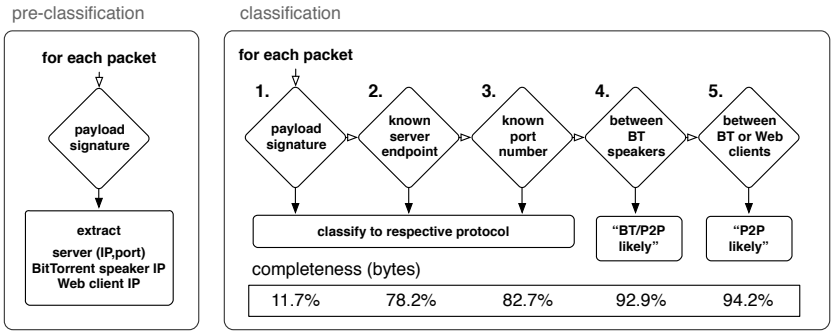


Fig. 2. Classification Pipeline.

II. Classification phase

We next process that same trace again and ensure that each packet proceeds through the classification pipeline shown in Figure 2. Once a packet can be attributed to an application, no further processing will be done for that packet.

Step 1: Payload signature matching. We match our previously extracted application signatures on each packet. Just by matching application signatures, we are able to classify 11.7% of the bytes exchanged at our IXP. This unexpected high number (recall that application signatures typically occur only in the first packets of a flow) is mainly the result of a proliferation of UDP-based BitTorrent data transfers, i.e., μTP [6]. μTP is a transport protocol based on UDP and includes its own header in every single packet. Thus, its classification is robust to sampling – in stark contrast to TCP traffic. The proliferation of μTP has also been reported in earlier studies [14, 22], as well as the rise of UDP-based applications using own headers in every packet [15]. In total, 11.3% of the packets matched a signature, of which 84.5% matched the BitTorrent UDP signature, another 11.7% matched an HTTP signature, 2.4% an SSL handshake on port 443, 0.94% a BitTorrent TCP signature, and 0.46% other signatures.

Step 2: Server endpoint matching. If a packet does not contain a valid application signature, we then check if the source or the destination (IP, port) tuple of the respective packet is a known server endpoint, as identified in our pre-classification step. If so, we classify the packet as belonging to the specific application. In this step, we classify 66.5% of bytes! This result highlights the efficiency of using a stateful application characterization approach. While we cannot sample application signatures on a per-flow basis, aggregating the information on a per (IP, port) endpoint basis largely overcomes the challenge posed by packet sampling. At the same time, we achieve a high confidence by relying on strong payload-based classification. This method works particularly well for popular client-server based applications, most prominently HTTP, where a large number of connections is destined to a comparably small number of server endpoints. To assess the impact of possibly stale endpoints (e.g., hosts that do not run the classified application on their server endpoint after some time), we repeated the classification by only using server endpoints that were identified within a time frame of 24 (12) hours, which reduced our completeness by only 1% (2%) of the bytes.

Step 3: Port-based classification. We next use a short list of 15 known port numbers (mapping to 13 applications) to classify respective packets as belonging to the cor-

responding application. In this step, we classify another 4.5% of all bytes. The largest contributor to this third step is RTMP (1.7%), for which no reliable signature is available. Interestingly, a significant fraction of traffic on port 1935 (RTMP) is HTTP traffic (and was thus already classified in the previous step), likely RTMP-inside-HTTP. Generally, we note that port-based classification can still be used reliably (but is not necessarily complete) when used in a conservative fashion, confirming prior studies [23]. For example, we observe that only less than 0.3% of the TCP traffic on port 80 did not match an endpoint which was detected using HTTP signatures (in the pre-classification). However, we find that more than 10% of the total HTTP traffic is not seen on port 80, and the most popular encountered non-standard ports are 8080 (3.8% of HTTP traffic), 1935 (2.9% of HTTP traffic) and 8000 (0.6% of HTTP traffic).

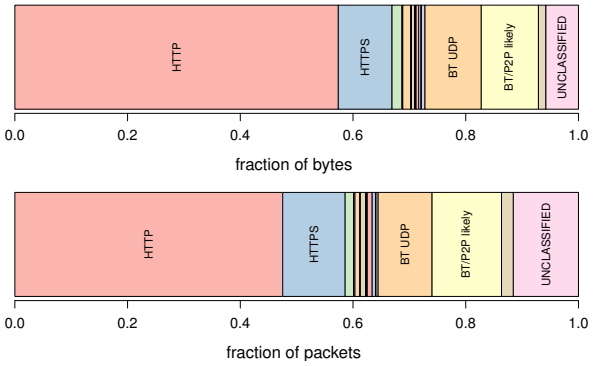
Step 4: Packet exchanged between BitTorrent speakers. In this step, we consider packets that were not classified in a prior step and classify them as "BT/P2P likely" if they are exchanged between two previously identified BitTorrent speakers. This step enables us to classify an additional 10.2% of the IXP's traffic. Depending on the individual client's configuration and capabilities, BitTorrent relies on TCP and UDP as transport protocol for data exchange as well as for exchanging control messages (e.g., DHT queries). While we are able to classify the bulk of BitTorrent UDP traffic (recall that we classified more than 11% of the traffic just using signatures), we are not able to classify the bulk of TCP traffic exchanged between BitTorrent speakers. In this step we account for this portion of the traffic. To provide further empirical support for this approach, we inspected partly sampled TCP messages of the peer-wire protocol [1] which corresponds to the transfer of *chunks*. By extrapolating the number of *piece* messages of the BitTorrent peer-wire protocol and multiplying it with the observed chunk size (16K in 99% of all cases), we can estimate that the pure content volume (excluding headers and control traffic) exchanged via BitTorrent TCP peer-wire connections is around 8%. Thus, we are convinced that the majority of the traffic classified in this step is indeed BitTorrent traffic. To acknowledge the lowered confidence and the possibility of other protocols contributing to this class, we classify these packets as "BT/P2P likely".

Step 5: Packet exchanged between Web clients or BitTorrent speakers. As a tie-breaking criteria, we classify all packets that are exchanged between either Web clients or BT speakers as "P2P likely". We only classify another 1.3% of the IXP's total traffic by using this heuristic. This small number suggests that most P2P likely traffic is indeed exchanged between BitTorrent speakers and was already classified in the previous step.

Using this classification approach, we are able to attribute 82.7% of the IXP's overall traffic directly to its corresponding application (Steps 1-3). More than 78% of the traffic can be classified either directly using payload signatures or by matching the packet to server endpoints identified using payload signatures – we only fall back to port-based classification for 4.5% of the traffic. Another 11.5% of the traffic is classified as "BT/P2P likely" using our heuristics based on the social behavior of hosts.

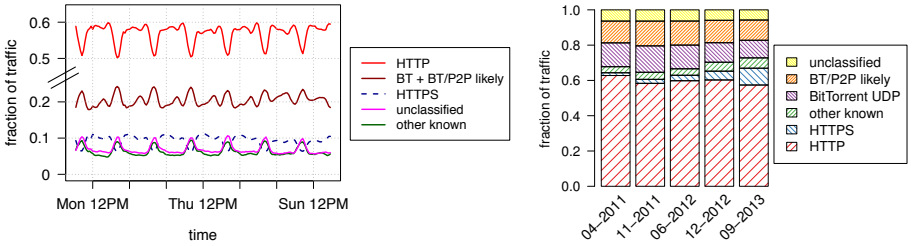
4 The Internet's Application Mix seen at an IXP

In this section, we discuss properties of the observed application mix. Figure 3 shows the result of our classification method when applied to the IXP's traffic, both in terms



Protocol	% bytes	% pkts
HTTP	57.39	47.52
HTTPS	9.53	11.08
RTMP	1.72	1.48
NNTP	1.41	0.87
NNTPS	0.63	0.38
SMTP	0.53	0.91
DNS	0.45	0.87
SSH	0.42	0.61
other known	0.68	0.74
BT UDP	10.00	9.57
BT/P2P likely	10.14	12.31
P2P likely	1.32	2.10
unclassified	5.78	11.56

Fig. 3. Application mix (September 2013) for packets and bytes.



(a) Weekly timeseries of the application mix.

(b) Evolution of the application mix over the last 2.5 years.

Fig. 4. Application mix over time.

of packets and bytes (flow statistics are not obtainable from our packet-sampled traces). We observe that HTTP(S) clearly dominates the application mix with a share of more than 65% of the bytes. While the increasing dominance of HTTP for a multitude of applications has been reported in prior studies (e.g., [26]), the other significant share of traffic is composed of the BitTorrent UDP and BT/P2P likely class, accounting for some 20% of the exchanged bytes. Other protocols such as email, newsgroups, RTMP etc. account for roughly 6% of the bytes exchanged at the IXP.

Figure 4(a) shows a timeseries of the contributions of the various applications for the 09-2013 trace. While we see that HTTP(S) always dominates (its share never drops below 55%), we observe a typical diurnal pattern indicating more pronounced HTTP(S) usage in the busy hour in the late afternoon. The share of BitTorrent/P2P peaks in the off-hours. Interestingly, we observe a second peak of BT/P2P activity each day, which is likely due to BitTorrent users in various time zones. Also the protocols in the "other known" category dominate in the off-hours. NNTP(S) is the largest contributor to this category and is reportedly used for file-sharing [23].

Next, we use five snapshots to infer the application mix as observed at this IXP during the last 2.5 years. The results for the exchanged bytes are shown in Figure 4(b). We observe that while the IXP's aggregate application mix is relatively stable, there is a significant increase of HTTPS traffic during these 2.5 years, from 1.9% in April 2011 to 11.1% in September 2013. Note that while in the snapshots from November

Study	Network Type	Method	Year	Bytes HTTP(S)	other known	BT/P2P	unclassified
[21]	5 large ISPs (peerings, Global)	payload-based	2009	52.1%	24%	18.3%	5.5%
[21]	110 Networks (peerings, Global)	port-based	2009	52%	10%	1%	37%
[23]	Large ISP (access, Europe)	payload-based	2009	57.6%	23.5%	13.5%	10.6%
[16]	Large ISP (backbone, US)	payload-based	2010	60%	28%	12%	N/A
[12]	260 Networks (peerings, Global)	port-based	2013	69.2%	4%	<7%	20%
[4]	Various (N/A, North America)	payload-based [5]	2014	≈ 70%	N/A	6%	N/A
[4]	Various (N/A, Europe)	payload-based [5]	2014	≈ 65%	N/A	15%	N/A
[4]	Various (N/A, Asia-Pacific)	payload-based [5]	2014	≈ 60%	N/A	30%	N/A
[4]	Various (N/A, Latin America)	payload-based [5]	2014	≈ 65%	N/A	9.4%	N/A

Table 2. Reported application mix in other studies (fixed, IPv4).

2011 to December 2012, both the share of HTTPS and HTTP traffic increased, there is a simultaneous decrease in HTTP and steep increase in HTTPS in 2013, suggesting a widespread switchover from HTTP to HTTPS in 2013.

5 The Application Mix: A Moving Target

5.1 The Aggregate View

The Internet's application mix has been the topic of numerous past studies by networking researchers and commercial companies alike. In the following, we report how the observed application mix at our IXP compares to other recent studies that not only relied on traffic data from different vantage points (and hence different types of peering links) but also used different application classification methods. Recall that in this study, we are only considering traffic that traverses the IXPs public peering links and have no visibility into the traffic that is sent over the private peering links established a this IXP. Table 2 lists some of the pertinent prior studies and provides information about the reported application mix, the type of traffic data used, and (where available) the classification method used.[2] A cursory comparison of the results of these studies with our findings suggests that the application mix of the Internet is rather homogeneous. That is, HTTP(S) dominates with a share of roughly 60%, no matter where in the network and with what methodology the application mix was measured. Other protocols such as BitTorrent or P2P seem to vary by region from around 10% to 30%, but these variations could also be in part due to varying classification approaches.

5.2 Beyond the Aggregate Application Mix

Next, we take a closer look at the apparent homogeneous nature of the Internet's application mix and examine in detail the application mix of the traffic that traverses the peering links of specific networks.

Figure 5 shows the application mix for each of the top-15 traffic-contributing member ASes of our IXP and top-3 traffic-contributing transit providers that are also IXP members. The type of the top-15 traffic-contributing IXP members is either *Content/CDN*, *Hoster/IaaS* or *Eyeball/Access*, and together they are responsible for 59% of the all the

[2] Note that the applications belonging to the "other known" traffic class vary across studies.

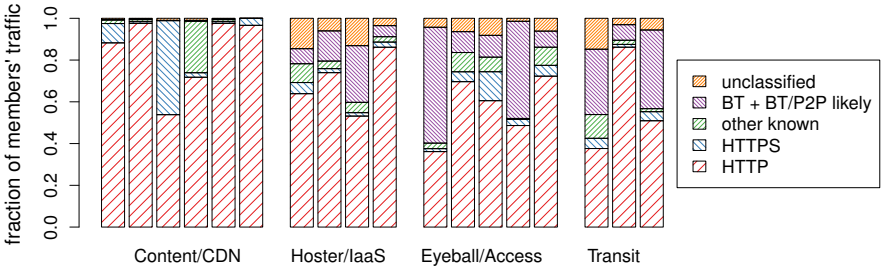


Fig. 5. Application mix of the top 15 traffic-contributing member ASes grouped by business type and the three most traffic-contributing transit ASes.

traffic (in bytes) seen at this IXP. We see that for all networks of type *Content/CDN* HTTP(S) traffic clearly dominates, with shares close to 100%. While most of these networks still rely mainly on HTTP, we notice one prominent network (third bar from the left) that has almost a 50/50 ratio of HTTP and HTTPS traffic. This example suggests that the earlier reported growth in HTTPS is mainly driven by some big content providers switching over to HTTPS. Overall, for networks of type *Content/CDN* we observe little or no application-mix heterogeneity on their individual links. Networks of the type *Hoster/IaaS* show a more diverse profile when it comes to their application mix. While HTTP still dominates, we see surprisingly no significant amount of HTTPS traffic. At the same time, these networks also see other types of traffic of various protocols as well as significant shares of BitTorrent traffic and unclassified traffic. Note that BitTorrent is also increasingly used to deliver video content or software [2]. In short, the diverse application mix contributed by Web hosters reflects the fact that they offer infrastructure services to a wide variety of companies and individuals, which in turn make different use of the provided resources. The results for *Eyeball/Access* networks show that the application mix of networks connecting end-users to the IXP also varies significantly. While for some of them, HTTP(S) (along with small fractions of other traffic such as email, RTMP, news) clearly dominates, we also see eyeball networks with more than 50% of BitTorrent traffic — the two networks with significant BitTorrent contributions are serving eastern European countries, while the other three networks are serving users in central Europe. This observation suggests that the differences in BitTorrent usage also reflect geographic properties (i.e., varying application popularity). The application mix seen for *Transit* networks is in general quite diverse as they typically carry traffic from a wide range of different networks.

The picture of the Internet's application mix sharpens even more when we look at the application mix seen on individual peering links. Figure 6 shows the application mix for the top-25 traffic-carrying bidirectional links at our IXP. The figure also includes the business types of the networks on either side of these peerings. Based on this set of links which see significant traffic, we observe a variety of different application mixes. While all Content-to-Eyeball links carry exclusively HTTP(S) and few other known applications, BitTorrent is the clear winner on two links between Eyeball networks. Thus, when taking into account the business types of two networks associated with a peering link, we notice a strong dependency on the resulting applications mix. The few links that show a more heterogeneous application mix are usually transit links or, interestingly, links involving Hosters and IaaS providers. When looking at the top-25

Fig. 6. Application mix of the top 25 traffic-carrying links.

unidirectional links (not shown), we see a similar pattern, where for Content-to-Eyeball links the resulting application homogeneity (i.e., HTTP) is even more dominant.

6 Conclusion

In this paper we developed a traffic classification methodology that is by and large able to overcome the challenges posed by packet-sampled traffic through the use of a *stateful* classification approach based on endpoint-aggregation. Using our new methodology we can attribute more than 78% of the bytes exchanged over the public switching infrastructure of a large IXP to their respective application by relying on strong payload-based classification. We attribute another 11.5% when including a heuristic based on communication patterns and classify an additional 4.5% using port-based classification. In the process, we observe that the aggregate application mix as seen at our IXP is largely consistent with that reported in other recent studies. However, when dissecting the traffic and examining the application mix of Internet traffic that traverses individual public peering links, we show that the application mix becomes heterogeneous but is strongly influenced by the business type of the networks on either side of a peering link.

Acknowledgements

We want to express our gratitude towards the IXP operators for their generous support and feedback. We thank the anonymous reviewers for their helpful feedback. Georgios Smaragdakis was supported by the EU Marie Curie IOF "CDN-H" (PEOPLE-628441).

References

1. BitTorrent Protocol Specification v 1.0. https://wiki.theory.org/BitTorrentSpecification.
2. Digital Trends article, October 12, 2013. http://www.digitaltrends.com/opinion/bittorrents-image-problem/.
3. L7-filter. http://l7-filter.sourceforge.net/.
4. Sandvine Global Internet Phenomena, 1H 2014. https://www.sandvine.com/downloads/general/global-internet-phenomena/.
5. Sandvine Traffic Classification. https://www.sandvine.com/technology/traffic-classification.html.

6. uTorrent Transport Protocol Specification. http://www.bittorrent.org/beps/bep_0029.html.
7. B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *ACM SIGCOMM*, 2012.
8. S. Alcock and R. Nelson. Libprotoident: Traffic Classification Using Lightweight Packet Inspection. Technical report, University of Waikato, 2012.
9. D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi. Tracking Down Skype Traffic. In *IEEE INFOCOM*, 2008.
10. A. Callado, C. Kamienski, G. Szabo, B. Gero J. and Kelner, S. Fernandes, and D. Sadok. A Survey on Internet Traffic Identification. *IEEE Comm. Surveys and Tutorials*, 2009.
11. V. Carela-Español, T. Bujlow, and P. Barlet-Ros. Is Our Ground-Truth for Traffic Classification Reliable? In *PAM*, 2014.
12. J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey. Measuring IPv6 Adoption. In *ACM SIGCOMM*, 2014.
13. A. Dainotti, A. Pescape, and K. Claffy. Issues and Future Directions in Traffic Classification. *IEEE Network Magazine*, 2012.
14. A. Finamore, M. Mellia, M. Meo, M. Munafo, and D. Rossi. Experiences of Internet traffic monitoring with Tstat. *Network, IEEE*, 25(3):8–14, 2011.
15. A. Finamore, M. Mellia, M. Meo, and D. Rossi. KISS: Stochastic Packet Inspection Classifier for UDP Traffic. *IEEE/ACM Trans. Networking*, 2010.
16. A. Gerber and R. Doverspike. Traffic Types and Growth in Backbone Networks. In *OFC/NFOEC*, 2011.
17. M. Iliofotou, B. Gallagher, T. Eliassi-Rad, G. Xie, and M. Faloutsos. Profiling-By-Association: A Resilient Traffic Profiling Solution for the Internet Backbone. In *ACM CoNEXT*, 2010.
18. T. Karagiannis, A. Broido, M. Faloutsos, and Kc claffy. Transport layer identification of P2P traffic. In *ACM IMC*, 2004.
19. T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. In *ACM SIGCOMM*, 2005.
20. H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K-Y. Lee. Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In *ACM CoNEXT*, 2008.
21. C. Labovitz, S. Lekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. In *ACM SIGCOMM*, 2010.
22. C. Lee, D. K. Lee, and S. Moon. Unmasking the Growing UDP Traffic in a Campus Network. In *PAM*, 2012.
23. G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *ACM IMC*, 2009.
24. A. W. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *PAM*, 2005.
25. T. T. T. Nguyen and G. Armitage. A Survey of Techniques for Internet Traffic Classification using Machine Learning. *IEEE Comm. Surveys and Tutorials*, 2008.
26. L. Popa, A. Ghodsi, and I. Stoica. HTTP as the narrow waist of the future Internet. In *ACM HotNets*, 2010.
27. P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger. Peering at Peerings: On the Role of IXP Route Servers. In *ACM IMC*, 2014.
28. InMon—sFlow. http://sflow.org/.
29. D. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia. Reviewing Traffic Classification. In *TMA*, 2013.
30. L. Wang and J. Kangasharju. Real-world sybil attacks in BitTorrent mainline DHT. In *IEEE GLOBECOM*, 2012.