TECHNISCHE UNIVERSITÄT BERLIN
FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIK
LEHRSTUHL FÜR INTELLIGENTE NETZE
UND MANAGEMENT VERTEILTER SYSTEME

Web content delivery, monetization, and search: Back-office and advertisement traffic on the Internet

vorgelegt von
Enric Pujol Gil (Dipl.-Ing.)
geboren in Barcelona, Spanien

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN
- DR. RER. NAT. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Jean-Pierre Seifert, Technische Universität Berlin
Gutachterin:	Prof. Anja Feldmann, Ph. D., Technische Universität Berlin
Gutachter:	Prof. Bruce MacDowell Maggs, Ph. D., Duke University
Gutachterin:	Konstantina Papagiannaki, Ph. D., Telefónica Research and Development

Tag der wissenschaftlichen Aussprache: 6. September 2016

Berlin 2017

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich diese Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum Enric Pujol Gil (Dipl.-Ing.)

Abstract

The World Wide Web has become the most used *information system* worldwide. It has fueled an unprecedented commercialization of the Internet by turning this *network system* designed for academic data exchange into a widely used social medium. After decades of operational experience with these two systems, researchers and engineers continue to be daily confronted with the urge to accommodate new users, scale up and deploy new Web services, and fulfill the users' quality expectations. This dissertation explores how these two systems influence each other. Namely, *i)* how the Web influences Internet's traffic dynamics, and *ii)* how the Internet's infrastructure and protocols impair Web usage.

We argue that there are three fundamental enablers for today's Web: *content delivery, monetization, and search*. On the one hand, *content delivery* entails the mechanisms that allow scaling up and serving Web content fast to end users worldwide. On the other, *content monetization* relates to obtaining the economic resources needed to deploy such infrastructure, create content, and sustain services. Finally, *content search* refers to the entire ecosystem that allows users to find resources on the Web almost in real-time without the need to navigate from one Web site to another Web site.

We study network traffic at multiple vantage points on the Internet, including a residential broadband network, backbone links of a tier-1 Internet Service Provider (ISP), two Internet eXchange Points (IXPs), and servers of a large Content Delivery Network (CDN). We show two distinct facets of Web traffic, for which we coin the terms *front-* and *back-office* Web traffic. The term *front-office* traffic refers to traffic exchanged between users and front-end servers. By contrast, the term *back-office* traffic designates traffic exchanged between two automated hosts (*machine-to-machine* traffic).

We analyze *front-office* Web traffic and more precisely *content monetization* via the advertisements that are displayed to the end users in an access network. We also study the prevalence of *adblockers*, as they can potentially disrupt the widely established business model of "free" content (one of the core elements on which the Web builds). We observe how the deployment of front-end servers has reduced latencies to many servers in the past years. By contrast, we note inflations of HTTP response times caused by back-end servers' activity related to *content delivery and monetization*. Hence, we devise a methodology to identify *back-office* Web traffic from data collected at the Internet's core. We find that this represents not only a significant fraction of today's Internet traffic but also today's Internet transactions. We further characterize it in the context of the three fundamental functions aforementioned, i.e., *crawling* (to find and index Web content), *real-time bidding* (to make advertisements more effective), and *request forwarding* (mainly to improve performance and reduce traffic).

In turn, the Internet's infrastructure and its operational protocols can alter the way users interact with the Web. We focus on two aspects thereof. First, we study the Internet's fundamental transition to a newer version of its *network-layer* protocol (IP), which affects the way users can reach the Web. In particular, we study *IPv6 usage* at a dual-stack ISP to reason about when and how users and Web content providers exchange data over IPv6 and the reasons that hamper its usage. Second, fueled by the *buffer bloat* debate, we investigate how buffer-sizing schemes and *transport protocols* like TCP influence QoE metrics for applications like Web browsing. To this end, we use data from a large CDN and report on the prevalence of excessive buffering in the wild.

Zusammenfassung

Das World Wide Web ist zu dem am häufigsten verwendeten Informationssystem der Welt geworden. Es hat die beispiellose Kommerzialisierung des Internets vorangetrieben und hat sich von einem Netzwerksystem, welches ursprünglich ein reines Forschungsnetz war, in ein weitverbreitetes soziales Medium verwandelt. Trotz jahrzehntelanger Erfahrung mit Netzwerken sind Forscher und Ingenieure tagtäglich herausgefordert, neue Nutzer in das Netzwerk aufzunehmen, neuartige Dienste bereitzustellen und die wachsenden Qualitätsansprüche der Anwender zufriedenzustellen.

Die vorliegende Dissertation fokussiert sich auf drei grundlegende Aspekte, welche das Web zu dem machen, was es heute ist: Inhaltsauslieferung, Monetarisierung und Websuche. Die Inhaltsauslieferung stellt Mechanismen bereit, welche eine schnelle Skalierung und die Auslieferung von Inhalten zu Benutzern weltweit ermöglicht. Die Monetarisierung der Inhalte liefert hingegen die wirtschaftlichen Ressourcen, um eine solche Infrastruktur aufzubauen, neue Inhalte zu generieren, und die Systeme zu warten. Die Websuche bezieht sich auf das gesamte Ökosystem welches es Nutzern erlaubt, Ressourcen in Echtzeit im Web zu finden, ohne dabei von Webseite zu Webseite navigieren zu müssen. Diese Dissertation untersucht die folgenden orthogonalen Aspekte: i) Datenverkehr im Internet, welcher der Inhaltsauslieferung, Monetarisierung und der Websuche dient, und ii) Kernaspekte der Applikations-, Transport-, und Netzwerkschicht des Internets, welche direkten Einfluss auf das World Wide Web haben.

Wir untersuchen den Datenverkehr von mehreren Internet-Aussichtspunkten aus, darunter ein Anschlussnetzwerk, Backboneverbindungen eines Tier-1 ISPs, zwei Internetknoten (IXPs) und Server eines großen CDNs. Hierbei betrachten wir zwei verschiedene Facetten des Web-Datenverkehrs und prägen die Begriffe des Front- und des Back-Office-Datenverkehrs. Der Begriff Front-Office-Datenverkehr bezieht sich auf den Datenverkehr zwischen Nutzern und Frontendservern. Im Gegensatz dazu bezeichnet der Begriff Back-Office-Datenverkehr den Verkehr zwischen zwei automatisierten Maschinen (Maschine-zu-Maschine-Verkehr).

Wir analysieren Front-Office-Webdatenverkehr und insbesondere Monetarisierung durch Werbeanzeigen, die Endverbrauchern in einem Anschlussnetzwerk angezeigt werden. Des Weiteren untersuchen wir die Prävalenz von Adblockern, da sie möglicherweise das etablierte Geschäftsmodell des "kostenlosen Inhalts" stören können, ein Kernelement des aktuellen Webs. Hier beobachten wir, dass der Einsatz von Frontendservern die Latenzen zwischen Endnutzern und Servern verringert hat. Im Gegensatz dazu haben sich insgesamt die HTTP Antwortzeiten erhöht, als Folge der Interaktionen zwischen Frontend- und Backendservern zur Inhaltsauslieferung und Monetarisierung. Wir entwickeln eine Methodik zur Identifikation von Back-Office-Datenverkehr und zeigen, dass dieser Datenverkehr nicht nur einen signifikanten Anteil des gesamten Datenvolumens, sondern auch einen signifikanten Anteil der heutigen Internettransaktionen ausmacht. Im Kontext der Websuche fokussieren wir unsere Arbeit auf den Anteil des durch "crawling" verursachten Datenverkehrs. Im Kontext der Monetarisierung betrachten wir "real-time bidding", um Werbeanzeigen effektiver zu schalten. Im Kontext der Inhaltsauslieferung analysieren wir "request forwarding", welches der Verbesserung der Leistung und der Reduktion des Datenverkehrs dient.

Front- und Back-Office-Webdatenverkehr werden jedoch auch von der Infrastruktur des Internets und den entsprechenden operativen Protokollen beeinflusst. Wir betrachten zwei Beispiele. Zunächst studieren wir die Transition des Internets hin zu einem neuen Netzwerkschichtprotokoll (IP), welches direkt beeinflusst, wie Endnutzer Inhalte aus dem Web beziehen. Insbesondere studieren wir die IPv6 Verwendung in einem dual-stack ISP um zu erörtern, wann und wie Inhalteanbieter ihre Webinhalte über IPv6 austauschen, und um die prävalenten Hindernisse aufzuzeigen. Im Kontext der Debatte über *buffer bloat* untersuchen wir, wie die Dimensionierung von Puffern und Transportprotokollen wie TCP die Quality-of-Experience (QoE) von Applikationen, wie dem World Wide Web, beeinflussen. Gleichzeitig verwenden wir Daten von einem großen CDN, um die Verbreitung von exzessiven Puffern im Internet aufzuzeigen.

Contents

List of publications

Parts of this thesis are based on the content included in the set of papers listed below. These papers have been peer reviewed and they have been co-authored with other researchers. All my co-authors are here acknowledged. I thank all of them for their valuable contribution.

International conferences

Enric Pujol, Philipp Richter, and Anja Feldmann. Understanding the share of IPv6 traffic in a dual-stack ISP. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (2017)

Enric Pujol, and Oliver Hohlfeld and Anja Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2015).

Enric Pujol, Philipp Richter, Balakrishnan Chandrasekaran, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs and Keung-Chi Ng. Back-office Web traffic on the Internet. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2014).

Oliver Hohlfeld, Enric Pujol, Florin Ciucu, Anja Feldmann and Paul Barford. A QoE perspective on sizing network buffers. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2014).

1

Introduction

The Internet has revolutionized the way the world communicates: it has become a worldwide accessible communication medium that allows individuals to interact independently of their geographical position and disseminate or broadcast information. Although recent estimates state that still only 40% of the World's population uses the Internet [38], many already consider access to this medium a basic human right [23]. After decades of operational experience with this network, a large number of researchers and engineers continue to be daily confronted with the urge to accommodate new users, scale up and deploy new services, and fulfill the users' quality expectations.

Whereas the dominant application on the Internet in 1972 was e-mail [47], the World Wide Web (WWW or the Web for short) takes this role today. As an *information system* accessible over the Internet, the Web has been the driving force for the unprecedented commercialization of this network of networks, turning a system designed for academic data exchange into a widely used social medium. Tim Berners-Lee and Robert Cailliau developed the World Wide Web in 1991. Shortly after the first Web browser and server software implementations appeared the number of Web servers on the Internet was approximately 50. At that time, Web traffic accounted for just 1% of the Internet's traffic [197]. By contrast, in 2010 the WWW was so popular that it alone contributed to 52% of the total traffic on the Internet [203].

The Web has revolutionized the way people publish, access, and search for content. In fact, some argue, it has also become the new "narrow waist" of the Internet (e.g., [245]), as manifested by the widespread and rapid adoption of browsers and Web-based technology. Indeed, the HTTP protocol provides a standard interface that many popular Internet applications rely on, including video, social networking, e-commerce, and software delivery. Despite being a "content-centric" protocol, HTTP provides many of the desired properties for new Internet architectures [245]. Consequently, HTTP –and by extension the Web– has become the de-facto method to link users and services distributed across computers around the world [197].

This dissertation investigates three fundamental enablers for today's Web: *content delivery, monetization, and search*. On the one hand, content delivery entails the mechanisms that allow scaling up and serving content fast to end users worldwide. On the other, *content monetization* relates to obtaining the economic resources needed to deploy such infrastructure, create content, and sustain services. Finally, *content search* refers to the entire ecosystem that allows users to find resources on the Web almost in real-time without having to navigate from Web site to Web site.

These three enablers have been a driving force for the popularization of the Web. Access to the Web at scale has, in turn, shaped the Internet itself. Today, millions of users scattered around the globe simultaneously request Web objects distributed across the Internet's infrastructure (who, at the same time, expect this system to reply fast). Reciprocally, the Internet's architecture and protocols can alter the way users interact with the Web, and by extension the realization of *content delivery, monetization, and search*. For example, Web users can employ technology that can block requests related to *monetization*, or content providers can deploy overlay networks to improve Web performance, i.e., *content delivery*. Thus, we posit that understanding today's Web involves analyzing how these functions manifest on Internet's traffic, as this is a crucial and necessary step to improve current and develop future Web-based services. Consequently, this dissertation explores the following orthogonal aspects: *i)* traffic on the Internet related to *Web content delivery, monetization, and search*, *ii)* how the Internet's infrastructure decisions impair Web usage.

1.1 Traffic that enables the Web

Web traffic on the Internet has been thoroughly studied since the introduction of the WWW; the majority of these studies taking a *user-centric* perspective, i.e., how end users consume Web content. We argue that "there is more to end-user interactions with the Web than is visible to end users", and, consequently, this dissertation explores the distinction between *front-office* and *back-office* Web traffic framed in the context of *content delivery, monetization, and search*. The first term, *front-office* traffic, refers to the traffic involving end users directly. In this case, *content monetization* is often realized with advertisements that are displayed to the end user. The second term, *back-office* Web traffic, applies to Web traffic exchanged between two automated hosts (e.g., back-end servers). In this dissertation, we focus on *back-office* Web traffic that realizes the functions mentioned earlier.

1.1.1 Back-office Web traffic: Content delivery, real-time bidding, and crawling

Although an end user typically views a Web page as a unit, recent studies demonstrate that a single Web page often contains links to objects that are delivered by a large and diverse set of servers [103, 180]. For example, a single Web page may involve several Web companies: parts of the Web page may be under the control of a content provider, a Web advertiser, a video streamer, a search engine, or a social network. Furthermore, fetching an individual part of a Web page may require communication between several of these servers. Today, when an end user requests Web content it involves not only the servers that receive HTTP requests from the end user's browser but also a whole service ecosystem consisting of proxies, content delivery networks (CDNs), ad-sellers, ad-bidders, back-end servers or databases, crawler bots, etc.

Note, not all back-office Web traffic traverses the public Internet. Some of it is carried over private backbones or within data centers. We also remark that our focus is HTTP(S) traffic, and hence acknowledge that other type of network traffic using different application-layer protocols may have the same objective.

Although traffic between Web servers and Web browsers is readily apparent to many knowledgeable end users, fewer are aware of the extent of automated Web traffic carried over the public Internet. This traffic, which may or may not triggered by end users, is essential for today's Web: it can relate to the three enablers for the Web mentioned earlier, i.e., *content delivery, monetization, and search*. The presence of this type of traffic and its implications for the Web ecosystem and by extension to end-users are not well understood. The reason being the difficulties to gain access to a vantage point where this kind of traffic is present.

1.1.2 Front-office Web traffic: Advertisements

Along with back-office Web traffic supporting the Web ecosystem, there exists also front-office Web traffic doing the same function, i.e., *content monetization* via advertisements displayed to the end user. Online advertisements (ads) allow content and services to be offered to users free of charge. The history of this business model goes back to the first clickable Web ads, which emerged around 1993 with the first commercial Web sites. *HotWired* was among the first to sell banner ads to companies such as AT&T and Coors [40]. The first central ad servers emerged in 1995 to enable the management, targeting, and tracking of users and online ads. The proliferation of ads on the Web started, which brought with it an increasingly complex infrastructure to serve these advertisements. DoubleClick introduced the process of online behavioral advertising in the late 1990s which used third party cookies to track users across Web sites and display ads based on their browsing patterns [290].

Today, advertising on the Web has become a broad and complex industry where entire exchanges trade user-specific information for the purpose of better ad placements [307]. This user specific profiling has raised many privacy and security concerns, in particular, among privacy and consumer advocacy groups. The debate has, for instance, led to agreements about the expiration dates of *cookies* as well as clear statements regarding the process of collecting user information [13]. While the Federal Trade Commission (FTC) identifies problems with online behavioral advertisement it allows advertisers to continue this practice with some safeguards such as greater transparency, provisions for consumers to opt out, and special handling of sensitive data, e.g., related to health and financial information [17, 18]. The motivation underlying the FTC's position is that the FTC views online advertisements as the key enabler for "free" content on the Web.

1.2 The Internet's influence on the Web

The Web and the Internet are in constant change. Hence, issues affecting one have immediate implications for the other. We next introduce three user and operational aspects present at the *application, transport and network* layers of the Internet that affect how users interact with the Web.

1.2.1 Application layer: Blocking of advertisements

Many Web users perceive ads as being not only invasive to their privacy but also annoying since they can distract them from the primary content they wish to consume, i.e., they degrade their Quality of Experience. This situation has resulted in a proliferation of tools to evade or block the ads by blocking ad-related HTTP requests or hiding the ads on the user's display. We refer to these tools as *ad-blockers*. Among the most used ad-blockers are extensions for Web browsers like *Adblock Plus* [4] and *Ghostery* [19]. According to Google and Mozilla usage statistics of browser add-ons, *Adblock Plus* is the most popular. More than 30M users surf the Web daily using a browser with this extension enabled [21, 27].

The use of ad-blockers is, however, perceived by the advertisement industry and content publishers as a growing threat to their business model [7,33]. Their rationale is that ad-blockers provide users a way to “evade paying” for the content they consume. Thus, some players try to put pressure on the developers of ad-blocking browser extensions to be excluded from the blocking. This exclusion is envisioned to be implemented by *i)* removing them from the filter lists, or *ii)* adding them to a whitelist [7]. Others try to detect ad-blockers and explicitly appeal these users to either whitelist their site or completely disable the ad-blocker [228]. Another claim regarding the rise of ad-blockers is that “as more end users adopt them, revenues decline, and the number of obnoxious advertisements increases” [3]. To break this vicious cycle, some players in this domain encourage advertisers to adopt “acceptable ads” as a compromise. The most important implication of this initiative is that advertisements that conform to these guidelines are by default whitelisted by *Adblock Plus* [6], a setting that end users can still override.

Given the impact of such technologies to *content monetization* and therefore to content producers, this dissertation explores the prevalence and adoption of ad-blockers by Internet users.

1.2.2 Transport layer: Queuing delays and buffer bloat

The advent of an Internet-based economy has fueled the growth of companies selling Internet-based services to end users worldwide. To them, service performance is indeed important as often such performance has a direct influence on their revenue [209]. For Internet Service Providers (ISPs) performance is a measure of the quality of their network and their interconnects to other networks. Similarly to telephone services, these ISPs offer Service Level Agreements (SLAs) to their customers via service level guarantees, i.e., Quality of Service (QoS). They use various metrics to measure QoS and ensure that they accomplish the requirements in the SLAs. The most common metrics in these SLAs are delay, packet loss, and jitter [9,20,29,202].

ISPs need to tune their networks often to satisfy the service level guarantees. Sometimes they apply traffic engineering techniques to influence routing and steer some of the traffic towards a destination over a specific path, usually using as criteria QoS metrics. Nonetheless, traffic engineering is not the only operation that allows ISPs to improve QoS metrics. One example is dimensioning packet buffers on their infrastructure: buffers can reduce packet loss rates in detriment of delay and vice-versa. Despite decades of focused research efforts, standards for sizing and configuring buffers in network systems remain controversial.

Recently, the buffer sizing debate has focused on the *existence* of large buffers in the network edge (buffer bloat [302]) and stimulated a discussion on its potential adverse effects. Excessive buffering *can* cause excessive queuing delays, e.g., in the order of seconds, in phases of congestion when the buffer capacity is fully utilized. Resulting excessive delays *can* degrade the performance from a users' perspective [302], e.g., by adversely affecting TCP due to increased round trip times or unnecessary timeouts. While empirical studies have shown the *existence* of large buffers, little is known on how often queues are *utilized* to degrade performance in practice. Also, the evaluation of the buffer bloat problem has so far focused on evaluating their influence on QoS metrics. In the absence of a solid understanding, the existence of large buffers on the Internet is currently used to drive engineering changes in Internet standards (see, e.g., [153]) and motivate new AQM approaches (e.g., CoDeL [232]).

Consequently, we argue that a deeper understanding of buffering effects on Quality of Experience (QoE) metrics is needed before altering important engineering choices affecting the Internet infrastructure.

1.2.3 Network layer: Transition to IPv6

In 1969 the first group of computers at four different universities started communicating with each other [47]. They formed the ARPANET, which many consider the progenitor of the Internet [207]. The standardization of the Internet Protocol (IP) began shortly after in 1970. IP is "the" *network* protocol that currently supports *connectivity* on the Internet. At that time, 32 bits seemed sufficient space for addressing hosts. Today, however, 32 bits are not enough. Address scarcity has become a problem because it affects the deployment of new services on the Internet. As a result, the Internet undergoes a transition of the IP protocol: from IPv4 (32 bits) to IPv6 (128 bits). However, the pace is slow, and much information on the Web is still not reachable over IPv6.

After almost two decades of IPv6 development and consequent efforts to promote its adoption, the current global share of IPv6 traffic remains low. Urged by the need to understand the reasons that slow down this transition, the research community has devoted much effort to characterize IPv6 adoption, i.e., *if* ISPs and content providers enable IPv6 connectivity. However, little is known about *how much* this is used in practice, i.e., which factors determine if two parties exchange data over IPv4 instead of IPv6.

Thus, we argue that gaining insights on *IPv6 usage* (i.e., the reasons that hamper IPv6 traffic) can help Web content providers to adapt their content-delivery strategy during the Internet's transition to IPv6.

1.3 Contributions

This dissertation makes the following contributions:

1. **Method to identify ad-related traffic in passively obtained network traces.** We devise a methodology to identify ad-related traffic in un-sampled full-packet traces. The method approximates structural Web site information and leverages open source code and filter lists of the most popular ad-blocker to date to realize the classification of HTTP requests.

   The business model of many popular Web sites relies on monetizing Web content via advertisements. At the same time, this ecosystem has raised concerns about end-users' privacy. Hence, this method enables the study the complexity of the ad ecosystem in the wild using as a vantage point a residential broadband network with 19.7K subscribers. We complement related work that *mostly* focused on analyzing this eco-system using active measurements.

2. **Assessment of ad-blocker usage in a residential broadband network.** Based on the results from the prior method, we propose two metrics to detect browsers using ad-blocking software and their configuration parameters.

   The ongoing debate on ad-blockers motives our study to gain insights about how end-users interact with advertisements. Our analysis on the same residential broadband network suggests that 22% of the most active users install an ad-blocker. However, most *Adblock Plus* users do neither subscribe to *EasyPrivacy* (the list that protects them from trackers) nor opt-out from the list of *non-intrusive* ads. We thus conclude that the main reason users install ad-blockers is to prevent annoying advertisements rather than protect their privacy.

3. **Method to identify Web traffic in sampled data.** We devise a methodology to identify Web traffic in high-volume fragmented data collected at the core of the Internet. Namely, we identify

Web end points from sFlow data (at two Internet eXchange Points), and packet sampled network traces (at long-haul links of a tier-1 ISP). The method combines active measurements (i.e., Internet-wide scans) with passive measurement techniques (i.e., payload signatures).

The characteristics of the traffic at the Internet's core infrastructure poses various challenges to monitoring systems designed to operate on access networks. First, due to the high rates of traffic, monitoring systems need to sample data, truncate packets, or both. Second, due to split-routing, flows are often incomplete. The proposed method circumvents these problems and allows us to reason about Web traffic at the core of the Internet.

4. **Characterization of *back-office* Web traffic on the Internet.** We leverage the previous methodology to illustrate the extent at which machines interact with each other on the Internet to sustain the Web, i.e., *content delivery, monetization, and search.*

   We show how a significant part of the Internet's Web traffic does not fit well into the classical understanding of Web traffic: one end-point is an end user and the other a Web server. We leverage multiple vantage points at the core of the Internet: two European IXPs with respectively 500 and 100 members, two long-haul links of a Tier-1 ISP, and server clusters of a large CDN. Our results suggest that *back-office* Web traffic represents not only a significant fraction of today's Internet traffic but also today's Internet transactions. Its volume contribution ranges on average from 10% to 30% per vantage point and can even exceed 40% for some time periods. We further characterize it and put it into the context of the three fundamental enablers aforementioned, i.e., *crawling* (to find and index Web content), *real-time bidding* (to make online advertisement more effective), and *request forwarding* (mainly used by CDNs to scale with demand and improve Web performance).

5. **Assessment of the Internet's transition to IPv6.** We investigate *IPv6-usage* in a dual-stack ISP and identify factors that limit the growth of IPv6 traffic.

   Transitioning services to IPv6 has implications for content providers and ISPs: it fundamentally affects the way users reach the Web, and, thereby, alters network traffic dynamics. Hence, it has the potential to affect Web performance as well. Using traces from a dual-stack ISP with 12.9K subscribers, we contribute to related work on *IPv6-adoption* with a study on *IPv6-usage* (when do they use IPv4 instead of IPv6 and vice-versa). We observe a strong intent for IPv6 traffic that IPv4-only content providers are not yet ready to correspond to, viz. users request content over IPv6, but content providers do not support IPv6. Due to dual-stack applications' preference for IPv6, dual-stack networks can experience a rapid and substantial increase of the IPv6 traffic share if a few major service providers enable IPv6.

6. **Use of QoE metrics to evaluate the performance implications of buffer-sizing choices.** We show how to use QoE metrics to assess the performance of network applications, e.g., the Web. Concretely, we evaluate the impact of buffer sizing choices on QoE metrics in two testbeds emulating *i)* an access network and *ii)* a backbone network.

   We inform the ongoing debate on *buffer bloat* with the following observations. Network workload, rather than the choice of a buffer size, appears to be the primary determinant of end-user QoE. As intuitively expected, sustainable congestion impacts both QoS and QoE metrics. Large (bloated) buffers further impair them. In the absence of congestion, we corroborate that even bloated buffers affect QoS metrics. However, our results suggest that the impact on QoE metrics is only marginal.

7. **Comments on Web performance.**

   We further contribute to *buffer bloat* debate with a passive measurement study. Using data from a large CDN (430 million randomly selected TCP/HTTP flows from 80M IPs in 235 countries) we investigate when are buffers over-sized and sustainably filled. Such conditions indeed occur in practice, as our empirical evaluation and other recent studies confirm, but their occurrence is relatively rare.

   We show how back-office Web activity has an impact on front-office Web traffic. We use data from a residential broadband network to show how in the past years more front-end servers are being deployed closer to end-users, indeed reducing RTTs. At the same time, we observe an inflation of some HTTP response times, due to *i)* front-end servers fetching content from other servers, and *ii)* real-time bidding for ad-related requests.

1.4 Structure of this dissertation

We organize this dissertation as follows. In Chapter §2, we discuss the related work about the topics addressed in this thesis. First, we present literature concerning the mechanisms and infrastructure that enable *content delivery, monetization, and search* on the Internet. Second, we discuss related work on the Internet topology and studies reporting on traffic dynamics. Finally, we also report on performance studies.

The first part of this dissertation relates to traffic that supports the Web. In Chapter §3, we investigate *front-office* Web traffic at a residential broadband network. Precisely, we study advertisement traffic and investigate the prevalence of ad-blockers in such network. Then, we characterize *back-office* Web traffic at various vantage points on the Internet in Chapter §4.

The second part focuses on how Internet infrastructure-related choices affect applications. In Chapter §5, we investigate how users in a dual-stack ISP use IPv4/IPv6 connectivity to reach the Web. Chapter §6 provides a QoE-centric study on the impact of buffer sizing choices for multiple applications, including the Web, VoIP, and IPTV. Moreover, we also include a passive measurement study on the prevalence of buffering in the wild.

We conclude this dissertation with Chapter §7, where we present a summary of this work and discuss future work.

2 Background

2.1 Enabling the World Wide Web

The World Wide Web (in short, the WWW or the Web) is an information system on the Internet that interconnects documents and other digital resources with *hypertext* links. The Web is inspired by ideas that Vanevar Bush proposed in 1945 [101]. In 1965 Ted Nelson coined the term *hypertext* to refer to non-sequential writing, i.e., a "representation of information based on a collection of linked nodes" [92,197]. Tim Berners-Lee et al. [93,94] introduced the Web as we know it today in the early 1990s, and, rapidly, it became the dominant application on the Internet [202,286]. The Web builds on the concept of *hypertext* to allow users access the information in a Web page (a node) and navigate to another Web site via the links contained in the former. The Web's application-layer protocol is the HyperText Transfer Protocol (HTTP), which is defined in various standards: HTTP/1.0 [95], HTTP/1.1 [144], and the most recent HTTP/2 [85].[1] As of today, the WWW contains information in a wide range of formats, including multimedia content such as video or audio. Moreover, its worldwide popularization is a driving force for the deployment of Internet infrastructure. Indeed, much of the Internet's server infrastructure is devoted to deliver Web content to end-users (*front-office* Web traffic). However, unnoticed to many users, many of these servers communicate with other servers in order to provide this service to them (*back-office* Web traffic). This observation motivates this section, where we describe the complex ecosystem of services that *enable* today's Web on the Internet. That is, how *Web content delivery, monetization and search* is currently realized on the Internet.

2.1.1 Content delivery and CDNs

In its most basic form, content delivery is implemented with a single Web server that answers requests from clients (end users). This simplistic model does not, however, scale with the current demand for content on the Internet. Leighton [206] identifies four main approaches to deliver content in today's Internet: *centralized hosting* (a small number of collocation sites), *big data centers CDNs* (a dozen of

[1] Hereafter we use term Web and HTTP traffic exchangeably.

high-capacity data centers connected to major backbones), *highly-distributed CDNs* (highly distributed infrastructure, including servers inside ISPs), and *peer-to-peer networks* (P2P). While small content providers can opt for the centralized hosting approach, as traffic increases managing the infrastructure becomes tedious. Large content providers with worldwide footprint often outsource content delivery to CDNs and shift their focus to their core business. The rationale for this decision is that CDNs can improve the end-user experience, react to flash crowds, mitigate attacks, and reduce the cost of content delivery using economies of scale [65, 206, 235]. The difference between the two CDN architectures aforementioned is that, according to Leighton [206], *highly-distributed* CDNs can better circumvent middle-mile bottlenecks and eliminate some problems that relate to peering, connectivity and routing. On the other hand, *big data center CDNs* typically involves lower maintenance and management overhead at the cost of more latency [177]. In the following, we provide an overview on how CDNs operate.

Deploying a commercial CDN. The fundamental operational principle of a CDN is based on deploying front-end servers (i.e., reverse proxies) close to the end users, as well as back-end servers in strategic locations. Back-end servers either host the content in data centers or are closer to the content origin server, depending on the CDN's deployment and operation strategy. If the front-end does not have a requested object available locally, it fetches the object from another front-end, a back-end, or the origin server. When the front-end proxies behave as clients they create back-office traffic (neither the client and the server are end users); when they act as servers, they can create either front- or back-office traffic. Hence, a CDN creates back-office traffic when none of the end-points IP addresses is an end-user (although the CDN may transport user-requested data). For a detailed description of various CDNs strategies we refer the reader to Akamai's platform [235,273], Google's backbone network [182], Microsoft Bing and AT&T anycast-based CDNs [70,146], and Netflix's technical description of peering requirements and appliances [28].

From caching static content to improving dynamic content delivery. Caching is an important and thoroughly studied technique to improve and scale up content delivery [80, 242, 295]. As stated by Davison [126], caches reduce network bandwidth usage, lessen user-perceived delays, and reduce the load on the origin servers. In other words, caching servers that are geographically closer to end-users than the corresponding *origin* servers can serve content at lower latency regimes as well as help reduce congestion at the core of the Internet. Thereby, caching is a fundamental technique for CDNs. The central assumption behind caching is that a significant fraction of requests involves previously requested objects (and that most of these do not change between requests) [132]. In 1997 Douglis et al. [132] analyzed traces from two corporations and found that 22% of the queried resources could have been cached (there were repetitions of the request). In 2011, Ihm et al. [180] analyzed the logs of the CDN CoDeeN. On the one hand, they observe that servers could cache 35%-54% of the objects and 15%-49% of the bytes. On the other, that 21-28% of the requests and 10-15% of the total bytes are not cacheable, mostly because many HTML and JavaScript objects are dynamically generated. Indeed, even for typical popular websites served by CDNs, the front-end hit rate is around 90% [273]. Hence, CDNs are challenged by the dynamic character of the modern Web as it involves forwarding requests to a remote origin. It is thus unsurprising that for a CDN being able to accelerate uncacheable content is equally critical to delivering strong QoE [206].

Reducing latency. One of the most important factors for Web performance is latency. The overall content delivery time has a direct impact on application performance and end user engagement [131, 201], as well as on revenue [97, 191, 209, 210]. For example, Linden [209] reports that every 100ms of latency could cost Amazon 1% of their sales and that every 0.5s extra delay in returning results page could drop Google traffic by 20%. In this content, Belshe [84] shows with controlled experiments that doubling bandwidth without reducing latency has a minimal effect on page loading times. Despite the importance of reducing latency, CDNs do not yet operate at the theoretical maximum speed [272]. However, CDNs deploy a number of optimizations for creating overlays to improve end-to-end performance. Leighton [206] provides a general set of optimizations related to transport, routing, pre-fetching, compression and delta encoding, and offloading operations to the edge. Some optimizations focus on the transport layer. Flach et al. [145] study the performance of billions of flows arriving at Google servers and argue that the round-trip time (RTT) and the number of round trips required between clients and servers largely determine the overall latency of most Web transfers. Therefore, they concentrate on improving completion times by addressing packet loss at TCP flows, i.e., *faster loss recovery methods*. Their work is motivated by the observation that 10% of the flows suffer from one packet loss at least and that these flows take on average five times longer to complete. Moreover, 77% of the losses undergo retransmission timeouts instead of a fast recovery. Sitaraman et al. [273] describe how to operate a caching overlay that can provide *speedups* between 1.7 to 4 across continents.[2] Krishnan et al. [200] observe that mapping to a geographically close server does not always result in the best performance due to *inflated* latencies. They use active measurement data obtained with the *ping* and *traceroute* tools and correlate it with flow records and BGP paths. They conclude that routing inefficiencies and packet queuing can "greatly undermine the potential improvements in RTT that a CDN can yield" [200]. While some of these queuing delays may be avoidable by tunning the CDN (e.g., by improving peering), others may be inevitable, e.g., due to buffer-bloat at the access network.

Avoiding inter-domain congestion. One of the optimizations proposed by Leighton [206] is to find "better routes" (i.e., faster and less congested routes). Consider, however, that network conditions affect content delivery differently, depending on the media that is being delivered to the end user. For example, the available bandwidth is more important to video-on-demand services than latency [131, 201]. This implies that there is less need to accelerate video delivery via overlays (less back-office traffic). Nevertheless, CDNs in general and video providers in particular may also generate a substantial amount of non-user-triggered back-office Web traffic when they pre-fetch content on their front-ends. One illustrative example is the video-on-demand provider Netflix, who accounts for 37% of the traffic in the US during peak time [52]. To improve service quality Netflix offers to deploy appliances within partnering ISPs and or establish direct peering at any of the multiple geographical locations they are present [28]. The costs associated with the infrastructure needed to support this amount of traffic has led to several peering disputes with ISPs [222]. If an ISP opts to deploy Netflix's appliances, it needs to consider that these will need to be updated with fresh content e.g., for 12 hours during off-peak hours at 2Gbps [28]. During this process they generate back-office traffic. Henceforth, back-office traffic may be user-triggered (e.g., forwarding requests to accelerate dynamic content distribution), or it may be also triggered by the CDN caching strategy (e.g., video pre-fetching).

Measuring CDNs. Much effort has been devoted to characterize CDNs. While most work has focused on characterizing CDN front-office traffic, some work has taken into consideration the impact of front-end and back-end communication. Huang et al. [177] compare a *highly distributed CDN* like Akamai against a *big data-center CDN* like Limelight. Starting from a performance study, they investigate each CDN infrastructure deployment strategy. For example, they use the *King* tool [165] to observe that Limelight DNS resolution delays are 23% higher than those of Akamai (for the 95^{th} percentile of them) or that the Akamai content server delay is 103ms in contrast to the 222ms of Limelight. In contrast to Akamai, Limelight uses *anycast* for its DNS system. At least 90% of the DNS requests are answered by servers at most 25ms away. These measures are, however, subject to various factors e.g., the number of data centers or user-to-server mapping strategies. Triukose et al. [291] conduct a similar study on Akamai performance and in particular on the quality of Akamai server selection. Adhikari et al. [63] perform an active measurement study to characterize the Youtube video delivery

[2] The ratio of the time to download the file directly from the origin compared to the time needed to download the file from the overlay.

system: they investigate Youtube use of *anycast* and *unicast* DNS name spaces in combination with HTTP redirections. One of their findings is that some servers may take some tens of milliseconds to start transmitting content. They most likely explanation for these delays is that these cache servers need to fetch the content from some back-end data center (back-office traffic). Chen et al. [113] conduct a similar study and investigate, via active measurements, the front-end back-end communication for two search engines i.e., Google and Bing. They observe that a "closer" front-end (in terms of RTT) does not necessarily imply better performance. In fact, they argue that front-end back-end communications may be affected by load fluctuations at the data center, the search algorithm in use, or the quality of the network connecting the servers. In other words, they argue that this —back-office— communication plays a critical role in the overall user-perceived performance.

2.1.2 Crawling and search engines

Search is one of the essential Internet Web services. Without search, the Internet would hardly be usable for most end users as their desired content would be difficult to locate. For example, using a three-day long dataset from a commercial ISP, Ben Houidi et al. [86] report that 11% of the URLs visited by users are due to search results from Google, one of the most popular search engines to date. Likewise, Kammenhuber et al. [184] report that the share of Google search operations and subsequent follow-up clicks in an educational network sums up to at least 6.8% of all transferred documents.

Search engines are build on top of three major services: crawling, indexing and searching [99]. Search relies on a back-end database which is typically populated by crawling the Internet and indexing the Web content found by the crawlers. For this purpose, search providers operate distributed server infrastructures that crawl the Web. Web crawlers (also known as crawl bots or spiders), are orchestrated to partition the Web and index different parts of it to more efficiently cover it [81]. Crawling and indexing involves requesting the Web page as well as following embedded links [99, 189]. Once the crawler bots have collected their data, they upload it to the search engine back-end infrastructure at large data centers, where it is processed with massively parallel indexing algorithms to enable fast search queries [81, 110]. When the end user receives the search results, they are typically sorted by relevance e.g., using algorithms like *PageRank* [237]. These results might be directly delivered to the end users, via overlays and/or the above mentioned CDNs [114].

Due to the size of the Web and its speed of change, effective crawling (e.g., index freshness) is a challenging task in itself [99, 109]. According to Dean et al. [127], Google clusters processed more than 20 petabytes of data per day in 2008. This data included: crawled documents, request logs or the set of most frequent queries in a given day. The volume of crawled data that was stored in Google back-end infrastructure by 2003 was more than 20 terabytes. Cho et al. [117] study how Web sites change over time. Overall, they find that while 50% of the sites changed within 50 days, 40 % of the Web sites change within a single week. Fetterly et al. [143] conduct a similar study and find that, usually, pages only change their markup. They also mention that Google crawls over 3 billion pages once a month to keep its index fresh. However, despite all engineering efforts devoted to retrieving and structuring content from the Web, a large portion of the Web remains un-indexed [215, 251], e.g., in the *Deep Web* (content that has not been yet indexed) or in the *Dark Web* (purposefully obfuscated content).

Along with the technical challenges imposed by the nature of the Web, there are numerous ethical implications regarding the use of Web crawlers [288]. For example, crawlers can impose a significant burden on Web servers, and badly-behaved crawlers could cause a denial of service [190]. Crawlers could also access content that content owners do not want to be indexed. One method to address this issues is based on a an exclusion protocol defined in the `robots.txt` file. Web masters can upload this file to their Web sites with guidelines for honoring Web crawlers. For example, which content is available for indexing [57], and the rate limits for the crawler agents [39]. Along with honoring these directives, best practices among the major search engines ensure that crawlers have appropriate reverse DNS entries along with well-specified user agents to avoid being blocked by Web sites.

Today's Internet search-engine landscape is concentrated around a small set of players i.e., Google, Microsoft Bing, Baidu and Yahoo [51, 58]. For many of these companies Web-search engines the primary source of revenue are the advertisements, i.e., those shown to users along with the search results Web page. In the case of Google, the most popular search engine worldwide, roughly 90% of its income is due advertising [256], and most of its revenue is due to ads included in search results [53]. This situation highlights that advertisements are fundamental not only for content publishers, as we next describe, but also for Web-search services and by extension for users relying on search engines to locate Web content on the Internet.

2.1.3 Monetization and the ad eco-system

To monetize their content, most Web sites rely on targeted online advertisements. In 2013, online advertising revenues in the United States were estimated to be $42.8 billion [24], an increase of 17% over the previous year. The increasing revenue stream of Web advertisement has given rise to another innovative part of the Web ecosystem: ad-sellers, ad-bidders, and ad-brokers—the ad-networks or exchanges [79, 307]. These parties negotiate placement of advertisements on today's Web. Advertisement exchanges consist of (i) publishers that sell advertisement space (ad space) on their Web pages, as well as (ii) advertisers that buy ad space on these Web pages. An ad exchange acts as a common platform to bring publishers and advertisers together. The matching between offered ad space on a Web site and interested advertisers is often performed using *real-time bidding* (RTB). Once an end user visits a Web page where ad space is available, the ad exchange auctioneer contacts the potential advertisers (i.e., the bidders), and provides information about the visitor to start a bidding process among the interested parties [72, 79, 276, 307].[3] Hence, a single visit of an end user to a Web page may trigger a larger number of connections in the background. Moreover, the final advertisement content is typically delivered via CDNs [72] which may in turn also generate more back-office requests in the process.

Content and services which are offered for free on the Internet are primarily monetized through online advertisement. A rich body of literature seeks to understand the scale, dynamics, mechanisms, economics and general interest concerns related to advertisements on the Web. In this section, we provide a summary of the most relevant lines of research concerning our work.

Privacy and security aspects. The first and largest line of research includes studies that investigate the extent to which advertisements violate or are in conflict with end users' privacy [164, 168, 194]. Other studies propose to address these issues with privacy-aware ad technologies [171]. Security-centric studies report the prevalence and properties of malicious ads [208, 308] and how to detect them [276, 277]. Another related study concerns ad-injecting browser extensions [306]. Other studies in this area paid particular attention to targeted advertisements. The goal of this type of ads is to improve the effectiveness of the displayed ads to the user by matching the users' interest. Farahat et al. [140] report an empirical evaluation and confirm the effectiveness of such advertisements. However, they also outlined ways in which more sophisticated targeting algorithms can cause harm. A large body of related work investigates privacy issues related to such user profiling, e.g., [125, 164, 168, 171, 194]. Gill et al. [157] conduct a passive measurement study to quantify the information that is aggregators collect and assess,

[3] The bidders may also contact other entities (i.e., *trackers*) to get information regarding the profile of the visitor [156, 307].

at the same time, the value of this data. Balebako et al. propose tools to mitigate behavioral advertisements [77]. Krishnamurthy and Wills [198] investigate privacy diffusion in a longitudinal study, where they report about the increasing aggregation of user-related information by a few companies, as well as the limitations of existing protection techniques.

Ad content and traffic characterization. Another notable line of research concerns the empirical characterization of the online advertisement landscape as well as its impact on Web site complexity. Guba et al. [163] and Barford et al. [79] characterize the ad-scape and highlight its underlying complexity. For instance, how ad exchanges enable real-time bidding to sell advert placements on a per user basis. Pujol et al. [248] report the scale of such "hidden" interactions in a passive measurement study. However, advertising has also a very visible and pronounced influence on today's Web site complexity. Krishnamurthy and Wills [199] report that 25–30% of the Web objects in the Alexa most popular sites are extraneous.

Mobile ads. There is one branch of related studies that focuses on advertising in the mobile domain. For example, Rodríguez et al. [292] quantify empirically mobile ad traffic via passive measurements. They reveal insights in the delivery mechanisms and outline means for optimized ad delivery. Targeting mechanisms used in the mobile world (e.g., location-based or user-based targeting) are not rare in today's Internet [98]. In this regard, Nath [230] characterizes the ads displayed in mobile applications and reports about the information that these apps collect for targeted advertising. Moreover, energy consumption is an important topic in the mobile domain. Thus, some studies also devoted their attention to quantify the energy consumed by mobile advertisements [67, 112, 124, 292]. In this regard, energy savings can be achieved by either pre-fetching ads [112], or by blocking them to reduce radio traffic [252].

Ad-blocking tech and tracking services. Butkiewicz et al. [102] report the share of extraneous content like ads in Web pages, and conclude that an ad-blocker can reduce the median number of requested objects per site by up to 75%. Guglemann et al. [162] investigate how to detect privacy-intrusive trackers and services from passive measurements. Kontaxis and Chew [192] describe a tracking protection mechanism for the Mozilla Firefox browser. Metwalley et al. [221] contribute to this line of research with a passive measurement study about the extent to which Web trackers follow users in a residential broadband network. One of their findings is that while many users install *Adblock Plus* (roughly 18% of the households), most of them do not install an tool to protect their privacy. Metwalley et al. extend this work and show that users with anti-tracking plugins still exchange data with trackers [220].

Online advertisement is a critical component of the Web, as it helps content providers to monetize their work and continue their activity. This traffic supports the Web, and its present both as front- as well as back-office Web traffic.

2.2 An overview on Internet's traffic and infrastructure

Internet topology. A very diffused mental model of the Internet topology formulated during its early days, described the Internet topology as a strong network hierarchy, mainly due to customer provider relationships. As Internet users began to consume content at a large scale, content providers began to search for direct connections to consumer networks with the objective of minimizing transit costs and deliver a better performance. The process is usually referred to as "the flattening of the Internet". One explanation for this phenomena is that providers, pressured by commercial competition, opt to peer directly with content consumers to *i)* reduce the costs imposed by using Tier-I networks and *ii)* improve latency, which translates into better QoE for end users [116]. In 2008, Gill et al. [155] corroborated via active measurements to/from `traceroute` servers, that many large content providers deployed their own wide-area networks, bypassing Tier-1 ISPs, to bring their content-delivery infrastructure closer to users. This phenomena is still prevalent nowadays as shown by Chiu et al. [116], who use `PlanetLab` nodes [31] and `RIPE Atlas` probes [34] to measure the number of hops between their probes and popular services on the Internet. They also observe that these paths tend to be shorter than random Internet paths: often a single hop.

Labovitz et al. [203] also corroborated this observation with a passive measurement study with data from 110 different networks. Based on their observations, they proposed a new and more dense model for the Internet topology. The model accounts with more interconnections among networks, and is divided into three strata: *i)* eyeball networks (customer IP networks), *ii)* regional and Tier-2 providers, and *iii)* the core of the Internet. Transit and national backbones form the core of the Internet together with "hyper-giants", e.g., large content providers, hosting providers and CDNs. Often, these hyper-giants are directly interconnected with customer networks and regional providers, or via an Internet eXchange Point (IXP).

Deployment of infrastructure. When content providers (CPs) establish new connections to other networks, they often extend the geographical coverage of their infrastructure. Lodhi et al. [213] report about the geographic expansion of content providers using public data that some operators willingly introduce into the `PeeringDB` database [30], These networks continue to increase their presence both at IXPs and private facilities. As the connectivity with other networks improves, content providers face the problem of mapping users to servers in an optimal fashion, e.g., to improve QoE or reduce costs. Many of them rely on DNS to assign users to servers, e.g., those that are geographically close to the user. A popular line of research focuses on characterizing how hyper-giants deploy infrastructure and interconnect to other networks. To circumvent the problem of deploying probes in every consumer network, Calder et al. [105] and Streibelt et al. [278] use the `EDNS-client-subnet` DNS extension [123], which enables to reverse engineer the DNS-based user-to-server mappings with a single vantage point. Calder et al. describe how Google increased their server-side infrastructure sevenfold in just a few months. For related work on CDNs front-end deployment, we refer the reader to §2.1.1.

Web traffic on the Internet. The flattening of the Internet has naturally influenced the traffic dynamics on the Internet. To date, most inter-domain traffic is exchanged between large content providers and eyeball networks [203]; a few Autonomous Systems (ASNs) contribute a significant share of the traffic, e.g., 30 ASNs to 30% of the traffic. A study by Cisco reports the global IP traffic increased more than fivefold from 2009 to 2014, and the estimates for the annual rate growth are at 23% for the period enclosing 2014 and 2019 [50]. In this report Cisco estimates that metro traffic will surpass long-haul traffic and will contribute to 66% of the total IP traffic by 2019. They attribute this change to Content Delivery Networks (CDNs), which will further carry half of the Internet traffic by the same year. Other recent studies [152,203,244] report that CDN traffic accounts for more than 50% of the total Web traffic. This percentage is expected to further increase in part due to the increasing traffic volume attributed to video delivery [131]. These studies imply that Web is, and will continue to be, the dominant application on the Internet. Other studies report the prevalence of this traffic at specific vantage points on the Internet, including:

- **Edge routers of multiple service providers:** Labovitz et al. [203] use data from a traffic monitoring system deployed at 110 Internet providers to report that Web (HTTP) traffic in 2009 contributed to 52% of the Internet Inter-domain traffic. Based on payload analysis, they further estimate that 10% of this traffic is actually video content. Czyz et al. [122] conduct a similar study with 260 networks and report that the ratio of Web traffic is roughly 70%.

- **The public switching infrastructure of IXPs:** Ager et al. [64] collect data at a large European IXP and report that the fraction of HTTP(S) traffic is 55% using as traffic classification method the port number. Richter et al. [254] use a payload signature-based approach to also classify traffic at a large European IXP. They report that Web traffic contributes to 67% of the traffic on that IXP.
- **Aggregation infrastructure of eyeball networks:** Maier et al. [216] study the traffic at a residential broadband network (RBN) and report that at least 57% of the traffic in the analyzed traces corresponds to Web traffic. The also dissect this network data per HTTP `Content-Type` and find that 25% of the HTTP bytes carry Flash video, followed by RAR archive files, which contributed to 14% of the HTTP traffic.

While the aforementioned studies confirm that Web traffic is strongly present in almost every vantage point on the Internet, they do not make a distinction between *front-* and *back-office* Web traffic. This dissertation takes a step further and provides an analysis of Web traffic that takes into account this difference and puts Web traffic in the context of *content delivery, monetization and search*.

2.2.1 Transition to IPv6

Claffy [119] argues that, given the pressure exerted on network operators by the Internet's transition to IPv6, there is a need to supply measurement data to properly inform technical, business, and policy decisions. In this context, some works have reported the IPv6 traffic share at multiple vantage points on the Internet. In 2008, most IPv6 traffic at a tier-1 ISP in the US was DNS and ICMP [186]. While initiatives such as the "World IPv6 day" in 2011 instigated the increase of IPv6 traffic at various vantage points [262], by 2013 the share of IPv6 traffic at European IXPs or at 260 network providers was still below 1% [122, 187, 254]. Nonetheless, every year IPv6 traffic experiences a many-fold increase [122]. Such development has encouraged studies on dual-stack networking performance [76, 118, 234, 241], active measurements of the Internet's IPv6 infrastructure [96, 214] and analyses of the AS-level topology [129, 158]. Moreover, a large body of literature has focused on measuring IPv6 adoption among ISPs and service providers [121, 122, 129, 158, 185, 186]. Some works seek to understand the root causes that slow down IPv6 adoption and find a slower pace of adoption at the edge compared to core networks [129], or poor IPv6 quality in the early days of this transition [233]. As of today, the IPv6 control and data planes are —when applicable— *almost* on par with IPv4 [211], while both control planes show signs of convergence [158]. In parallel to the research community, standardization bodies have invested decades to address IPv6-related aspects. Relevant to our work are fall-back mechanisms for dual-stack applications [304] (*happy eyeballs*) and their implementations (see e.g., [60, 178, 179, 264]). This dissertation complements this body of work with a passive measurement study at a dual-stack ISP to shed light on why some data exchanges occur on IPv4 instead of on IPv6.

2.2.2 Queueing delays

Congestion on Internet links can lead to a degradation of QoS metrics e.g., high packet loss. In this context, packet buffers are deployed in network devices to reduce the packet loss caused by transient traffic bursts. Hence, a buffer works as a "knob" that trades packet loss for queuing delay (latency). The effect of packet buffers highly depends on the characteristics of the traffic traversing the device where the buffer is deployed. Dischinger et al. [130] observe that many DSL links exhibit queue lengths higher than 130 ms.[4] Specifically, they find that some uplink queues may add delays in the order of

[4] The recommended maximum end-to-end latency for interactive applications is set by the ITU G.114 standard to 150 ms.

seconds. Kreibich et al. [196] argue that over-sized buffers are endemic in access devices. In particular, they observe buffers that can introduce over 250ms (1s) delays for 8Mbps downlink (1Mbps uplink) capacity. Allman [69] argues that these studies provide evidence that bufferbloat *can happen*, but not if it *does happen*. In this study, conducted in a FTTH network with 90 homes for 14 months, Allman concludes that the magnitude of full and bloated buffers is modest i.e., 50% (94%) of the samples are bounded by 100 ms (250 ms) queuing delays. Chirichella and Rossi [115] exploit the BitTorrent protocol to study queuing delays that affect remote hosts. They, as well, conclude that most peers (99%) suffer a delay below 1 s. Martin et al. [218] study cable-modems and observe that that upstream traffic may experience delays in the order of seconds in the presence of congestion. Jian et al. [169] investigate buffering in the mobile domain.

These aforementioned studies fueled the recent bufferbloat debate [154,302] regarding a potential degradation in Quality of Service (QoS). Indeed, prior work has shown that buffer sizing impact QoS metrics. Examples include aspects such as per-flow throughput [246], flow-completion times [205], link utilizations [83], packet loss rates [83], and fairness [296]. Sommers *et al.* studied buffer sizing from an operational perspective by addressing their impact on service level agreements [274]. However, QoS metrics and even SLAs do not necessarily reflect the actual implications for the end user.

2.3 Performance evaluation of networked applications

Le Callet et al. [48] provide a detailed description of the relationship between performance, *Quality-of-Service* (QoS), and *Quality-of-Experience* (QoE) metrics, which we here summarize. The ITU defines QoS as the “totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.” [46].[5] Tanenbaum and Wetherall [286] argue that bandwidth, delay, jitter, and loss define the needs of a network flow.[6] All together these metrics determine the QoS (Quality of Service) required by a network flow and its corresponding application. For example, Web browsing has low jitter and medium bandwidth, delay and loss requirements. In turn, Le Callet et al. [48] define QoE as “the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user’s personality and current state”. They argue that while QoS metrics focus on performance aspects of physical systems, QoE deals with how users assess the performance of a system.[7] However, they conclude that, in many situations, QoE is highly dependent on QoS. Given these two different perspectives, we discuss related work on these topics in the following sections.

2.3.1 A QoS perspective

Due to the tight relationship between QoS and QoE metrics, it is unsurprising that service providers use, despite their limitations, QoS metrics as key performance indicators. In fact, much effort is devoted to improve QoS metrics (e.g., by placing caches close to users to reduce latency and avoid inter-domain congestion). Latency is the QoS metric that has triggered most attention in the past years. We refer the reader to Briscoe et al. [100] for a detailed survey of state-of-the-art techniques to reduce latency on the

[5] International Telecommunication Union (ITU)

[6] A set of packets grouped by certain fields e.g., source and destination IP addresses.

[7] Möller provides the following definition for *performance*: “The ability of a unit to provide the function it has been designed for.” [223].

Internet. QoS degradations may occur at the *middle mile*, at the *first mile*, or at the *last mile*.[8] Thus, much work has focused on understanding the characteristics of network links on the Internet.

A common belief is that the *last mile* is the current bottleneck on the Internet because it entails most of the links with lowest bandwidth and highest latency. Hence, most related work tries to characterize this segment of the Internet. More recently, the *middle mile* has also become a concern among operators and researchers alike, as peering disputes have lead to congestion at inter-connects and thereby impaired the quality of services (see, e.g., [222]).

Active measurements and platforms. Most of the related literature seeks to understand and characterize the *last mile* using active measurements. Dischinger et al. [130] use the DNS to obtain a list of 1.8K hosts in 11 major commercial cable and DSL providers in the US and study them using active measurements from the outside (e.g., from an educational network). Sundaresan et al. [283] rely on measurements from 4.2K `SamKnows` probes [35] installed within the customer premises of 8 different ISPs in the US. The Broadband Internet Service Benchmark project (`BISmark`) seeks to deploy home routers capable of realizing measurements [32, 282]. Instead of replacing the home router, the `RIPE Atlas` platform provides probes that users can install at their premises. Researchers around the world can use this platform to conduct measurements [34]. To date, this project accounts with 9K operative probes and 25K users. Related work provides advise on how to use this platform [75], and highlights potential sources of interference when using it [176]. Other approaches seek to leverage software instead of hardware to conduct active measurements from the customer premises. Sánchez et al. [259,260] propose to scale up measurements at the edge by developing a framework deployed as an extension for a BitTorrent client (a P2P network). More than 90K users installed their extension `Dasu`. Finally, Oana and Teixeira [159] report that some home gateways may introduce biases affecting the bandwidth estimation results of state-of-the-art tools. A less extense set of works studies the *middle mile* and the *first mile*. Chandrasekaran et al. [74] report on the path characteristics the paths between servers of a CDN by using `traceroute`-based measurements. Clark et al. [120] design a technique tailored to expose points of congestion across between ISPs. Their main finding is that links are, usually, adequately provisioned. Paths exhibiting congestion are very specific, mostly involving sources of high volume like Google or Netflix.

User-generated data. A different line of research relies on user-generated data to investigate access performance. Kreibich et al. [196] develop a Java applet called `Netalyzr`, which users access via their Web browsers and which allows them to perform a wide range of measurements e.g., unveil hidden caches, DNS manipulations, or the capacity of access buffers. They provide insights based on a dataset of 130K sessions recorded in 2009, from 6.8K organizations in 186 countries. A discussion on how and why users perform such measurements is available in [195]. Similarly, Canadi et al. [107] use a data set collected over six months at a Web site providing speed tests on demand [37]. Their data set consists of 54 million tests from 59 metropolitan markets. Bauer et al. [82] provide a detailed analysis about some of the limitations and caveats that are employed to assess the quality of broadband networks.

Passive measurements. Maier et al. [216] conduct a passive measurement analysis on a Residential Broadband network with roughly 20K customers and report on the impact of the access technology on metrics such as RTT or the achieved throughput. Sargent and Allman [261] study performance at a fiber-to-the-home (FFTH) network. Concerning the *middle mile*, Feamster [141] reveals that the aggregate utilization across the interconnects of seven different ISPs is usually low, even during peak periods ($\leq$ 50%). Although this observation highlights that congestion is many times not a problem, the study also shows that some links are in contrast very congested.

[8] Peering and transit links form the Internet's *middle mile*, i.e., the Internet infrastructure where networks exchange data [206]. The *first mile* refers to the origin infrastructure. The last mile entails the infrastructure that connects the end-user premises with the first IP router of the ISP.

2.3.2 A QoE perspective

Assessing human quality perception is challenging due to its subjective nature. While network performance is typically expressed by QoS metrics, QoS and QoE are fundamentally different concepts that can influence each other; QoS represents a *network-centric* view whereas QoE is rather a *user-centric* view (see Figure 2.1). QoE depends on a multidimensional perceptual space that includes *i)* system influence factors (e.g., QoS measures, transport protocols, or device specific parameters), *ii)* human factors (e.g., mood, personality traits, or expectations), and *iii)* contextual factors (e.g., location, task, or costs). For an extensive discussion of factors influencing QoE, we refer to [48]. These features are not necessarily independent of each other and do not always have clear mappings, e.g., users tend to give different opinion scores for the same stimulus, e.g., depending on mood, expectation, and memory. While some QoE influence factors include QoS metrics (e.g., packet loss), QoE depends on a larger set of influence factors that cannot be derived from QoS metrics alone and requires a new set of metrics. In fact, there exist ongoing work within standardization bodies to define QoE metrics for applications like VoIP, Video, or Web browsing. These metrics are rooted in psychological tests that involved human subjects in the metric *construction phase*. In the *application phase*, however, they allow automatic quality assessments without user involvement; i.e., conclusions on user-experience can be drawn from testbed evaluations *without costly user involvement*. This is an appealing property as it enables the automatic exploration of a large state space that involves a significant number of different scenarios in *controlled experiments*.

The Web browsing experience (Web QoE) can be quantified by two main indicators [135]. The first indicator is the *Page Loading Time* (PLT), which is defined as the difference between a Web page request time and the completion time of rendering the Web page in a browser. The second indicator is the time to the first visual sign of progress. The PLT is often used as a *proxy* metric to measure QoE. As argued by Möller and Raake [225], the PLT is a key factor impacting QoE. Moreover, it can be measured without the need of obtaining users feedback and has the advantage that there exists an ITU QoE model (i.e., G.1030 [45]) that can be used to map PLTs to user scores. The Web QoE does not directly depend on packet loss artifacts, but rather on the completion time of underlying TCP flows. Thus, factoring in various network conditions—which influence the TCP performance—is particularly relevant for understanding Web QoE from a network only perspective. Given that the PLT as measured in a browser can be approximated from flow completion times as parameter, the PLT is sometimes considered as a QoS parameter. However, since the applied G.1030 model logarithmically maps PLT to QoE, it can be mis-believed QoS parameters can (always) be mapped to QoE. We therefore note that other QoE models are of higher complexity as the take different input parameters, which cannot be directly derived from a QoS parameters, e.g., speech signals.

Page loading times. Many factors influence the PLTs. One of them is the characteristics of the access network. Sundaresan et al. [285] observe that latency becomes more important as the bandwidth increases. In particular, they find that PLTs do not decrease after reaching 16 Mbit/s of available bandwidth. Moreover, they show that the latency of the last mile can contribute up to 23% of the time until the client receives the first byte of a Web page. They propose a set of optimizations for home routers that can reduce PLTs up to 53%. Similarly, Singla et al. [272] observe that the median PLT for popular Web sites is 34× longer than the theoretical minimum (the speed of light). However, the PLT is not only affected by network conditions but also by the structural design of Web pages. Wang et al. [298] present a profiler to investigate dependencies within the page load process. They conduct a study on 350 pages to unveil that computation contributes to 35% of the critical path. Based on this observation, they suggest that not only network but also computation time should be taken into consideration when aiming to optimize PLTs. They also highlight that synchronous JavaScript objects may block the process parsing HTML. In the context of CDNs, they find that reductions in the page load times are not

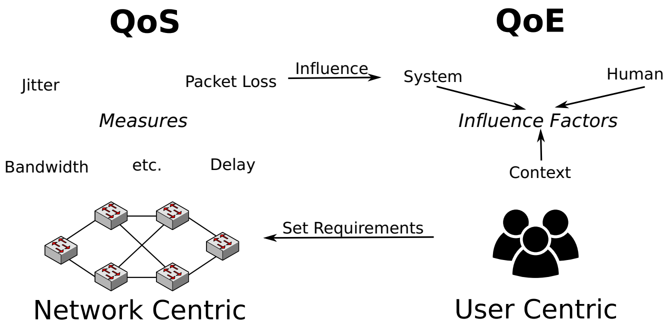


Figure 2.1: Conceptual difference between QoS and QoE

proportional to the cached bytes. Wang et al. [297] highlight some of the inefficiencies in the page load process and propose a proxy-based system to mitigate them. They find that ¾ of the CSS resources are not necessary for the initial phase of the rendering process, or that 15% of the PLT is wasted on waiting parsing-blocking resources. The underlying application protocols also influence PLT. Naylor et al. [231] use PhantomJS to show that using HTTPS significantly increases the PLT, e.g., on fiber HTTPS can add more than 500 ms extra delay to 40% of the pages in their study. Likewise, Wang et al. [299] show that while SPDY can reduce up to 7% for some specific scenarios, such benefits are less pronounced in the face of object dependencies and computation.

Summary

The popularization and commercialization of the Internet and the Web has resulted in a large and complex ecosystem of services and infrastructure devoted to provide immediate access to Web content to millions of users around the world. Many users are not aware of this ecosystem's complexity and how it "enables" the Web for them: it realizes *Web content delivery, monetization and search*. While these enablers influence the Internet's infrastructure and traffic dynamics, the Internet infrastructure can impair Web usage in turn. Hence, in the first part of this dissertation, we study *front-* and *back-office* Web traffic using passive network traces obtained at various vantage points on the Internet. The second part of this dissertation focuses on how Internet infrastructure influences the Web. In particular, we first study the Internet connectivity, i.e., how users use IPv6 to reach the Web. Second, we shed light on the impact of buffering-sizing schemes on QoE metrics.

Part I

Enabling the World Wide Web: Impact on Internet's traffic

3 Front-office Web traffic: Advertisements

Content and services which are offered for free on the Internet are primarily monetized through online advertisements. This business model relies on the implicit agreement between content providers and users where viewing ads is the price for the “free” content. Hence, an important part of the traffic that is delivered to an end-user (front-office traffic) directly supports other content in the Web.

This status quo is not acceptable to all users, as manifested by the rise of ad-blocking plugins which are available for all popular Web browsers. Indeed, ad-blockers have the potential to substantially disrupt the widely established business model of “free” content—currently one of the core elements on which the Web is built.

In this chapter, we shed light on how users interact with ads. We show how to leverage the functionality of *Adblock Plus*, one of the most popular ad-blockers to identify ad traffic from passive network measurements. We complement previous work (see §2.1.3), which mainly focuses on active measurements, by characterizing ad-traffic in the wild, i.e., as seen in a residential broadband network of a major European ISP. Finally, we also assess the prevalence of ad-blockers in this vantage point. The contributions of this chapter are the following:

1. We present a methodology to extract structural Web site information from packet header traces. With this methodology at hand, we can use the *Adblock Plus* functionality to classify ad traffic in passive measurements. We validate the effectiveness of this methodology with an active measurement study based on different configurations of an instrumented browser, e.g., running *Adblock Plus*.
2. We leverage two indicators to infer *Adblock Plus* usage: *i)* low ratio of ad requests, and *ii)* connections to the *Adblock Plus* servers. Based on these metrics we estimate that 22% of the most active users likely use this extension. Moreover, we have indications that, perhaps, most *Adblock Plus* users do neither subscribe to *EasyPrivacy* (the list that protects them from trackers), nor opt out from the list of *non-intrusive* ads.
3. We find that advertisement traffic contributes to a significant percentage of requests, roughly 18% in both of the studied traces. We dissect ad requests by type and find that 11% of them match the list of *non-intrusive ads*, while the rest is distributed among advertisers and trackers. Given

the prevalence of this traffic in terms of requests, we also characterize various aspects thereof, including object types and server-side infrastructure.

4. We show that back-end functionality can be inferred from front-office traffic characteristics. In particular, we detect latency inflations in the HTTP response times that we attribute to content-delivery and real-time bidding.

The remainder of this chapter is structured as follows: we first discuss the basic functionality of *Adblock Plus* and the corresponding filter lists in §3.1. We then describe in §3.2 our measurement methodology and its subsequent evaluation. Section §3.3 summarizes the two data sets as well as the involved vantage points. We present our first results about *Adblock Plus* usage in §3.4. We then elaborate on traffic- and infrastructure-centric aspects of advertisements in §3.5. In §3.6 we show the effects of back-office activity on front-office Web traffic. We discuss the limitations of our methodology and present our conclusions in §3.7.

3.1 Background

There is a wide range of browser extensions available to end-users who want to evade or protect themselves from the *ad-scape*. *Adblock Plus* is arguably among the currently most popular ad-blockers [26]. Users can configure this browser extension to *i)* block or hide advertisements, and to *ii)* protect their privacy by blocking trackers. Another popular tool is *Ghostery*, which mainly focuses on protecting end-users' privacy. The Electronic Frontier Foundation's *Privacy Badger* [15] shares the same objective; but in contrast to *Ghostery*, which is proprietary software, *Privacy Badger* is open source—and based on *Adblock Plus*. Another option is the *NoScript* plugin which is designed to interactively disable executable Web content such as JavaScript, which is often used by advertisers and trackers.

To assess the popularity of these extensions we can refer to statistics reported by popular browsers or to recent work by Metwalley et al. [221]. They report, relying on a passive measurement study, that 80% of the households visible at their vantage point do not use any of these popular plugins. Among those that do, *Adblock Plus* dominates, i.e., *Adblock Plus* is installed at 10%-18% of the households. Indeed, less than 3% of the households exhibit evidence of other installed plugins. Given the popularity of *Adblock Plus*, we next describe how this particular extension operates.

Adblock Plus

At the heart of *Adblock Plus* is the mechanism to filter adverts based on filter rules that appear in filter lists. The regular expressions that form the set of filter rules follow a specific syntax, which is described in detail in [5, 41]. If a filter rule matches a URL that is not otherwise white-listed, *Adblock Plus* will prevent the Web browser from requesting the URL. Hence, this extension reduces network traffic as it averts undesired ad-related objects from being fetched. However, some Web pages embed advertisements into the (main) HTML document, e.g., textual advertisements. Since this document is required to render the page, the tool does not block the download of these HTML documents. However, *Adblock Plus* includes functionality to hide such —otherwise displayed— embedded ads via `CSS` modification. Note that these ads are still transferred over the network even though they will not be displayed in the browser to the user.

Adblock Plus users can obtain various lists of filter rules via a subscription mechanism. There are several filter lists available for different purposes. When an end user installs the *Adblock Plus* extension for the first time, the plugin subscribes itself to two filter lists. The first one is named EasyList and its goal is to remove ads from English Web pages. The second list is called non-intrusive advertisements (acceptable ads) and its purpose is to white list the advertisements that are blacklisted by EasyList but comply with the directives summarized in [6]. This list and, in particular, its activation by default is most likely the origin of the controversy described in [7] (i.e., advertisers paying to be white-listed). Note that users may still opt to deactivate this list with a single click. There are additional lists to which *Adblock Plus* users can subscribe. Examples include *i)* customizations of EasyList to non-English Web pages, or ii) EasyPrivacy, which aims to protect the end users' privacy by blocking Web trackers.

Measurement challenges. At first glance, given that this ad-blocker only uses filter lists, it should be easy to emulate its behavior on a passively collected network trace. However, this is not that simple as *Adblock Plus* relies on the information contained in the DOM tree of the Web site to classify Web elements as adverts. For example, the classification can be based on whether an image is displayed in an *iframe*, which cannot be detected by inspecting the URL. Thus, this ad-blocker relies on the entire structure of a Web page rather than purely the URLs. To reconstruct the entire structure of the Web page one needs parse the payload part of the traces. For privacy reasons we cannot and do not have access to this part. Our traces only include HTTP header information. Hence, our methodology has to rely on the information available in the HTTP headers. How we tackle this challenge is discussed in the next section. Our methodology enables us to approximately reconstruct the Web page meta-data from HTTP headers.

3.2 Methodology

In this section we describe our approach for identifying ad-related traffic, i.e., we need to devise a methodology for identifying Web objects and separating them into ad-related or non ad-related objects. This is where we rely on *Adblock Plus*, as this plugin itself includes an engine capable of performing this classification. We rely on *Adblock Plus* functionality because it is the most popular ad-blocker to date. The main requirement for our methodology is that it has to classify ad traffic in TCP/HTTP header traces captured via passive measurements. Thus, instead of using a *Adblock Plus* browser directly we use *libadblockplus* [25], a C++ *wrapper* around *Adblock Plus*, available as part of the project. This *wrapper* allows us to classify URLs (Web objects) from the traces in an off-line fashion without the need to operate a full browser. Figure 3.1 shows a sketch for the classification methodology.

3.2.1 Identification of ad-related traffic

We use *Bro* [239] HTTP analyzer to extract information about all HTTP transactions in network traces. This information includes *i)* both the `Host` and the `URI` fields in the request header, *ii)* the `referer` field in the request, *iii)* the `Content-Type` field in the response, and *iv)* the `Content-Length` field present in HTTP responses. We also extended the *Bro* analyzer to parse and include *v)* the `Location` header field present in response headers that relate to HTTP redirections. Processing passive traces with *Bro* gives us a list of Web objects. We then invoke *libadblockplus* to classify each of these objects into ad and non-ad objects. However, *libadblockplus* cannot properly classify a URL as an advert using only the information that is self-contained in the URL's string. Many rules in the filter lists apply to specific combinations of domains, i.e., a Web object hosted in a specific (advertiser) domain from a specific (publisher) domain. Thus, *libadblockplus* requires the following information to properly classify a URL: *i)* the requested URL itself, *ii)* the rest of URLs in the Web page that triggered the request that is currently processed, and *iii)* the type of the content that is being requested e.g., `document`, `script`, `stylesheet`, `image`, `media` or `object`. As mentioned above, the *Adblock Plus* browser extension

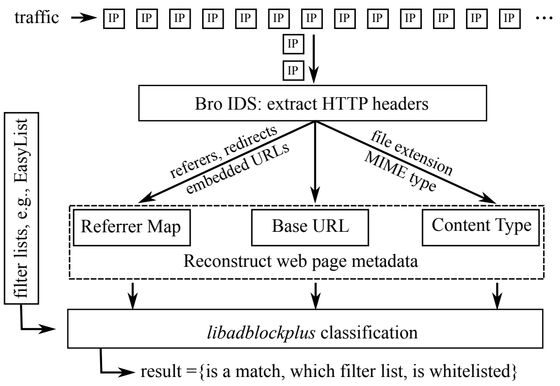

traffic
IP
Bro IDS: extract HTTP headers
referers, redirects
embedded URLs
file extension
MIME type
filter lists, e.g., EasyList
Referrer Map
Base URL
Content Type
Reconstruct web page metadata
libadblockplus classification
result ={is a match, which filter list, is whitelisted}

Figure 3.1: Approach to classify ad requests.

relies on information contained in the DOM tree. We, in contrast, have only the information available in the HTTP headers. However, we can still use this information to obtain a *partial view* of the relationships between the Web objects in a Web page. We tackle this challenge in the following way. See the middle boxes in Figure 3.1 for an illustration of the approach.

1. **Construct a "Referrer Map".** First, we extract the set of related URLs for a given request. To this end we construct a *referrer map* that approximates the set of URLs in a Web page based on the chain of observed HTTP referrers. Our approach is based on the *StreamStructure* and *ReSurf* methods discussed in [180,305]. We also use these methods to obtain the Web site that triggered the request. We construct the referrer map out of the values in the `referer` header fields in the requests. However, there are a few cases in which this chain may be "broken". One typical example is when the request following a redirection to a new URL has no `referer`. This is the reason why we extend *Bro* to also parse the `Location` response headers. With this small modification we can add this type of missing referrers to the map of referrers. Furthermore, we also insert the URLs that we can find embedded within the URL of a request into the referrer map.
2. **Infer the "Content Type".** Second, we infer the type of the content in the following way: one of the pitfalls in HTTP traffic analysis are mis-matches between the `Content-Type` of the request and the actual content. Schneider et al. [265] showed that while mis-matches often occur due to the format (e.g., `jpeg` vs. `png`), they actually agree on the general category (e.g., `image`). In these cases the mismatch does not impair our classification because *libadblockplus* relies on general categories. For other mismatches we parse the URL to map the following file extensions to content types: i) *.png*, *.gif*, *.jpg*, *.svg*, *.ico* (`image`) ii) *.css* (`stylesheet`) iii) *.js* (`script`) iv) *.mp4* and *.avi* (`media`). As a rule of thumb, we rely on the `Content-Type` field when the file extension does not yield a type. Some redirections may lead to mis-classifications. For instance, there are cases where a URL within an `<img>` HTML tag results in a redirection. Suppose there exists an exception filter for that URL and its content type is `image`. To *Adblock Plus* this request is an image, since it can glean this information from the tag. We, on the other hand, would filter it because we do not have access to this information. Here, the referrer map helps us to set the appropriate content type for the URL that is being redirected by inspecting the type of the consequent request.
3. **Infer the "Base URL".** Third, we process the URLs to avoid conflicts with the filter lists. Namely, we noticed that *libadblockplus* mis-classifies some requests because they include parts of the URL of a previous request in the query string. While the *Adblock Plus* plugin does not filter the second request, *libadblockplus* does. To prevent this type of mis-classifications we normalize

> the query strings by removing dynamic values. However, there are some filters in these lists that specify values for the fields in the query strings of a URL, e.g., *@@*jsp?callback=aslHandleAds**, where * represents a wild-card. If we would normalize this string to *jsp?callback=X*, it would not match anymore the previous *exception filter* and thereby we would mis-classify the corresponding URL. Hence, we take care not to overwrite the values in the query strings that appear in the filter lists.

Finally, recall that *Adblock Plus* also includes functionality to block text advertisements that are included in the HTML itself. Quite a number of HTML documents embed these advertisements, which means that the browser extension will not block the associated request because blocking it would also imply blocking non-ad content. Instead, the browser plugin hides this ad content during the rendering phase of the page, in a process called *element hiding*. Since we cannot and do not have access the packet payload we cannot parse the Web page's content and thereby, can neither detect nor comment on this type of advertisements.

3.2.2 Metrics to detect ad-blocker usage

To identify if an end user has installed an ad-blocker we rely on the following observation: a browser with such an extension should issue less ad requests than a browser without any ad-blocker. In other words, a low number of ad requests is a strong indicator for the presence of an ad-blocker.

Hence, we have to identify all requests of a particular end user and compute the ratio of ad requests. The ratio for an end user depends on *i)* the browser configuration, i.e., whether there is an ad-blocker installed and, if so which ad-blocker with which configuration, and *ii)* which sites the user visits. Given that many of the most popular sites do indeed have extraneous content [79, 102, 199], the likelihood that an active user visits such a site is substantial. Thus, we can use an active measurement study (see §3.2.3) to find ad-ratio thresholds that distinguish between ad-blocker and non ad-blocker users.

Although our primary goal is to identify *any* ad-blocker user, we can additionally rely on plugin updates to identify specific ad-blockers like *Adblock Plus* or *Ghostery*. For instance, the former plugin regularly checks for updates of the filter lists to which the user has subscribed. The update frequency is driven by the expiration time specific to each list, e.g., *EasyList* has a soft expiration date of 4 days [1] and *EasyPrivacy* soft-expires already after a single day [2]. In fact, the *Adblock Plus* contact frequency is quite high: typically upon browser bootstrap or once per day [221]. Hence, monitoring connections to *Adblock Plus* servers is a good indicator for the presence of *Adblock Plus*. To identify *Adblock Plus* servers in the traces we rely on multiple DNS resolvers to obtain an up-to-date list of *Adblock Plus* server IPs.

3.2.3 Method evaluation and validation

We complement our passive analysis with an active measurement study that has two goals *i)* validate that our methodology can classify ad traffic, and *ii)* identify ad-ratio thresholds to differentiate between end users that browse the Web with an ad-blocker and those who do not. Accordingly, we instruct a popular Web browser to fetch Web sites using different configuration modes, e.g., with *Adblock Plus* enabled or disabled. In parallel we capture the browsers network traffic with *tcpdump* and apply the above methodology to the passive traces.

Active measurement setup

We instrument the widely used browser Chromium with Selenium [36] to crawl the Alexa top 1000 sites. We run the browser in a virtual frame-buffer on a dedicated GNU/Linux machine connected to a university campus network. For each URL in the Alexa top list we start a new browser instance with an empty cache, wait 5 seconds before starting a *tcpdump* traffic trace, load the URL and wait another 5 seconds before closing the browser and *tcpdump*. For each URL we repeat this process 7 times, once for each of the following browser profiles:

- **Vanilla:** We do not activate any plugin.
- **AdBP-{Ads|Privacy|Paranoia}:** We activate the *Adblock Plus* plugin and configure three separate profiles using the following lists i) the *EasyList* and *non-intrusive advertisements* (Ads), ii) *EasyPrivacy* (Privacy), and iii) *EasyList* and *EasyPrivacy* (Paranoia).
- **Ghostery-{Ads|Privacy|Paranoia}:** We activate the *Ghostery* plugin and configure three separate profiles, which block the following object categories i) Advertisements (Ads), ii) Privacy (Privacy), and iii) all categories, including Analytics, Beacons and social media widgets (Paranoia).

This experiment results in seven sets of passive traces for each of the top-1000 Alexa Web sites corresponding to the seven different browser profiles viz., with and without *Adblock Plus* and/or *Ghostery* enabled. With this information—the configuration of the Web browser that produced the network trace—we can apply our methodology to each set of traces and, thereby, assess the accuracy of our methodology.

Impact of ad-blockers on traffic

As described in §3.2, we use *Bro* HTTP analyzer to extract the HTTP information from the set of traces collected during the experiment. We then classify the requests using *libadblockplus*. Table 3.1 summarizes the total number of HTTP(S) requests in the traces and the corresponding classification of requests according to the filters of *EasyList* and *EasyPrivacy*.

Our first observation is that ad-blockers indeed significantly reduce the number of HTTP and HTTPS requests. For instance, in the *AdBP-Paranoia* mode the browser issues 9K less HTTP requests than in the *Vanilla* mode. The number of HTTP requests issued by a browser configured with *Adblock Plus* in the most aggressive mode is roughly 80% of the corresponding value for the *Vanilla* mode. In this context, 6% of the ad requests in the trace for the *Vanilla* mode either match a filter in *EasyList* (8.1%) or one in *EasyPrivacy* (8.3%). These observations are consistent with those reported in related work, e.g., see [102, 199]. The number of HTTPS connections follows likewise this trend, i.e., 2.9K connections less. This implies that browsers and servers also exchange ad traffic over HTTPS, a case not covered by our methodology.

The number of requests for both plugins configured in the *Paranoia* mode also differs. We remark that the numbers reported in Table 3.1 are subject to configurations and do not strictly reflect the actual filtering performance of each plugin; rather, they reflect the existence of the filtering process. Nonetheless, Table 3.1 shows that the number of objects classified as ad requests is a strong indicator for the presence of an ad-blocker. As expected, ad-blockers hinder many requests from being issued and thus, the number of identified ads with our methodology is small (indicated in bold numbers in the table). In the absence of an ad-blocker, the number of ad requests is significantly larger (see the vanilla browser configuration row).

Browser Mode	**#HTTPS**	**#HTTP**	**#EL$_{hits}$**	**#EP$_{hits}$**
Vanilla	7,263	57,862	4,738	4,807
AdBP-Pa	4,287	48,599	**6** *	**6** *
AdBP-Ad	5,254	53,435	**10** *	4,279
AdBP-Pr	5,189	55,717	3,627	**7** *
Ghostery-Pa	2,908	48,765	940	624
Ghostery-Ad	5,734	57,425	1,326	4,668
Ghostery-Pr	6,902	55,394	4,514	2,865

Table 3.1: Active measurements: Aggregate results for the Alexa top 1K list. Browser modes include Paranoia (Pa), Ad-blocker (Ad) and Privacy (Pr). Classification of URLs based EasyList (EL) and EasyPrivacy (EP). Ad-blockers lessen the total number of requests and lower the ratio of ad requests.

We note that our approach mis-classifies a small number of requests. We indicate *false positives* with a $*$ in Table 3.1. *False positives* are requests that the *Adblock Plus* browser plugin does not block but our methodology classifies as ad-related objects. We manually investigate these cases and offer the following explanation: some are due to inconsistent `Content-Type` values in the responses. For instance, in one case *Bro* reports the MIME *text/x-c*. Our methodology maps this request to the content type `object`, but a manual inspection of the object reveals that this object is in reality a JavaScript object. Remapping the content type to `script` triggers an exception filter that prevents the mis-classification. In fact, the main source of mis-classifications are URLs to JavaScript objects where the `Content-Type` field is set to *text/html*. Modern browsers circumvent this problem since they infer the type of the object without relying on the HTTP headers. Our methodology has to rely on the HTTP headers and thereby is affected by such inconsistencies.

Identifying ad-blocker users

Table 3.1 illustrates the effectiveness of our approach for classifying ad requests in a network trace in a fashion similar to an ad-blocker. An important question relates to the ratio of ad to non-ad requests, and in particular, which values are useful indicators to infer ad-blocker usage. To answer this question we show in Figure 3.2 a series of box-plots for the ratio of ad requests across the following browser configurations: *Vanilla*, *AdBP-Pa* and the *Ghostery-Pa* configuration modes. For each of these three modes we execute three experiments. We randomly select 1, 5 and 10 sites of the Alexa top 1K list, and compute three ratios of ad requests. Our motivation for choosing different number of sites is to represent users with different levels of activity. We repeat this 1K times. When comparing the box plots we can see that the ad-ratios differ significantly if the number of page loads is sufficiently large, i.e., when users are active. Accordingly, we use a discrimination threshold of 5% when the number of requests is sufficiently large, e.g., 10 page loads or 1K requests. Using a slightly higher or lower threshold does not alter the results significantly.

3.2.4 Limitations

Our methodology leverages *Adblock Plus* functionality to provide insights into how ad-blockers may influence ad traffic dynamics and thereby, Web traffic. Although we show that the proposed methodology is capable of classifying ad requests in HTTP header traces, the classification approach comes at a price.

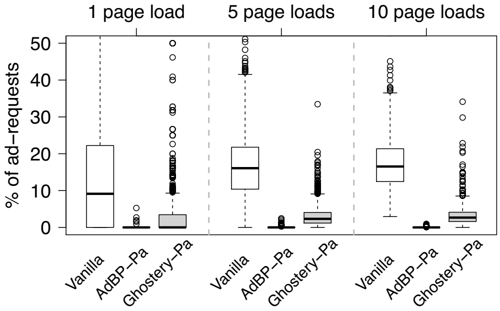


Figure 3.2: Active measurements: ratio of ad-requests per browser configuration. Comparison among 1K iterations of 1, 5, 10 randomly selected page loads. The presence of an ad-blocker is more evident when the user becomes more active.

First of all, the classification of ad requests in header traces is complicated by the lack of structural information in the absence of the HTML payload; individual HTTP requests cannot be associated with Web objects (e.g., if an image is embedded in an iframe). This structural information is leveraged by *Adblock Plus* to improve the ad detection. To tackle this challenge, we propose a methodology to partially reconstruct the Web page structure. Our methodology mainly employs the construction of a referrer map which associates individual requests (e.g., images, video, CSS, or JavaScript) with the accessed page. While this approach allows us to cluster related requests by page, it cannot reconstruct the entire Web page structure as needed by *Adblock Plus* to achieve higher detection accuracy. We suggest that a complete reconstruction is only possible by accessing the payload and executing the embedded JavaScript code, which might further manipulate the Web page structure. Moreover, ad and non-ad objects can be transferred over HTTPS, or a mixture of HTTP and HTTPS (e.g., the landing page over HTTPS and the ads over HTTP). Since URLs in HTTPS transfers cannot be analyzed, we cannot always associate all requested objects with a page when constructing the referrer map of such a page. We are, thus, also unable to reason about the prevalence of ad traffic carried over HTTPS connections.

Second of all, our methodology can underestimate the volume of ad traffic in the presence of *hidden ads*, i.e., ads embedded in the main HTML of the page whose retrieval cannot be blocked. To identify these kind of ads we would require access to the entire HTML document (i.e., packet payload), which is not possible in our study. In cases where the payload can be analyzed, our methodology can be extended to detect *hidden ads* and address the challenges discussed above. Lastly, we remark that there are many other ad-blockers available to end users besides *Adblock Plus*. We may not be able to detect users with different extensions (or even *Adblock Plus* with custom configurations) if they are tailored to block different objects than those affected by *EasyList*.

Furthermore, analyzing the number of ad requests (indicator 1, see §3.4.2) involves a set of potential biases: *i)* custom configurations and different filter lists (e.g., Web site whitelisting), *ii)* cascaded effects in which the blocked ad request triggers subsequent requests, or *iii)* the interaction of *Adblock Plus* with other blocking extensions. The immediate consequence of such biases is that we may *underestimate* the overall usage of ad blocking browser extensions. Moreover, caches and ad blocking middle-boxes or proxies can also decrease the number of observed ad requests. Consequently, confusing *Adblock Plus* instances with ad blocking proxies will lead to *overestimation* of the number of *Adblock Plus* users. To limit the effect of these biases, we introduced the filter list download indicator (§3.4.2) which monitors updates triggered by *Adblock Plus*. However, since such updates are fetched via HTTPS, we cannot observe the *User-Agent* string required to differentiate different browsers behind a NAT. We point out that this is a general limitation of the presented *results*, which we produced using the lists mentioned in

Section §3.1, rather than of the applied *methodology*. We selected *Adblock Plus* based on its popularity among end users and note that our approach can be extended to consider other ad-blockers.

3.3 Vantage point and datasets

We had access to two anonymized traces from a Residential Broadband Network (RBN) of a large European ISP. These two traces were collected within two customer aggregation networks in the same city. The first trace was captured at a low level which carries the traffic of about 7.5K DSL customers to the Internet. The second trace was captured at the next higher level, which carries the traffic of about 19.7K DSL customers. The up-link speeds are 3 and 10 Gbps, respectively. Table 3.2 reports the dates when the traces were collected.

The monitoring infrastructure uses Endace DAG network monitoring cards [16]. These cards support a port-based classification, which is appropriate for HTTP(S) traffic (see [216, 254]). Hence, HTTP traffic can be associated to TCP traffic from (or to) port 80. Likewise, HTTPS traffic relates to *EasyList* downloads can be associated to TCP traffic from (or to) port 443 from (or to) IPs in the list of *Adblock Plus* servers that host these lists. We obtained this list with active measurements before and after the trace was captured. They did not exhibit differences.

Like many other ISPs, most of the customers of this ISP have a home gateway at their premises that performs Network Address Translation (NAT). These gateways multiplex many user devices, and thereby also browsers, to a single IP address. To report the prevalence of ad-blockers in residential broadband networks we have to identify unique devices and more specifically Web browsers. Maier et al. [217] showed that a good indicator for separating HTTP traffic from multiple devices behind the same NAT is the HTTP `User-Agent` strings of the browsers (in contrast to the strings used, e.g., by software update tools or media players). The rationale behind it is that the `User-Agent` string includes information about the operating system, browser version, etc. Thus, we use the pair end-host `IP` and `User-Agent` to separate our data by end device. Given that most ISPs assign addresses to their customers dynamically, we can only associate an IP address to a household for traces with short duration. Table 3.2 gives an overview of the two traces collected in this ISP. The first data set corresponds to a 4-day long trace of the smallest set of customers for which we only capture HTTP traffic, i.e., 7.5K . The second trace is a shorter trace, but it captures peak time traffic for a larger number of end users, i.e., 19.7K . In the remainder of this work we use the latter trace to elaborate on ad-blocker usage, and the former to describe general characteristics of ad traffic.

We pay careful attention to respect and preserve the privacy of end users in our study. First and foremost, we process, aggregate, and analyze the data on a private and secured infrastructure. Second, the IP addresses of the end users are anonymized at the time of the packet capture, i.e., the real IP addresses of the end users were never stored to disk and are unknown to us. Third, we automate the ad classification process which, when completed, truncates every URL in the logs to a fully qualified domain name (FQDN), thereby removing sensitive information.

3.4 Prevalence of ad-blocking technology

Using the methodology from §3.2 we can proceed to assess ad-blocker usage in the residential broadband network trace `RBN-2`. To infer if an end user is using an ad-blocker, we use the following two

Trace	**RBN-1**	**RBN-2**
Date	11^{th} Apr. 2015	11^{th} Aug. 2015
Time	00:00	15:30
Duration	4 days	15 and ½ hours
Subscribers	7.5K	19.7K
HTTP$_{bytes}$	18.8T	11.4T
HTTP$_{reqs}$	131.95M	85.09M

Table 3.2: Dataset overview

indicators: i.e., *i)* the ratio of ad to non-ad requests, and *ii)* automatic *EasyList* downloads by *Adblock Plus* (see §3.2.2).[1]

To this end, we have first to identify browsers in the trace. Along with browsers, residential broadband networks indeed manifest a rich and complex mix of other HTTP-based applications. To illustrate this mix, we show in Figure 3.3(a) the total number of retrieved HTTP objects vs. the number of retrieved ads for each `IP` address and `User-Agent` pair on a log-log scale.[2,3] Due to the large number of data points, we use a heat map to capture the density within each area of the plot. We find in total 508.7K tuples of `IP` and `User-Agent` in `RBN-2`. We see the whole range: some pairs only account for a few requests while others issue orders of magnitude more requests. Overall, we observe 18.89% ad requests in this data set.

Most relevant for us is that there is a substantial number of pairs that request many Web objects but hardly any ads. These are the ones in the lower right hand side of Figure 3.3(a). They are most likely browsers that have an ad-blocker installed or visit only sites without advertisements. However, there are many more points in this plot than we would have expected given the number of monitored DSL-lines (19.7K different households); more than 25 different `User-Agent` strings per household in average. Upon closer manual inspection we find pairs that identify devices like consoles and Smart TVs. On the other hand, given that most modern end-user devices run many HTTP-based applications in parallel, we also find pairs that correspond to desktop-based gaming applications or mobile apps (these applications use custom `User-Agent` strings). Since advertisements typically appear within Web sites and mobile applications, we can discard `User-Agent` strings that do not correspond to this type of applications. Moreover, since in-app ads differ substantially from browser ads, we here limit our analysis to Web browsers. Namely, we restrict our analysis to sessions for which we can associate the `User-Agent` either to a well-known desktop browser or to a mobile device browser. Indeed, a manual inspection of the pairs shows that some of the points on the right hand side of Figure 3.3(a) correspond to mobile apps, which we do not want to consider. Thus, we next use the `User-Agent` strings to identify popular Web browsers.

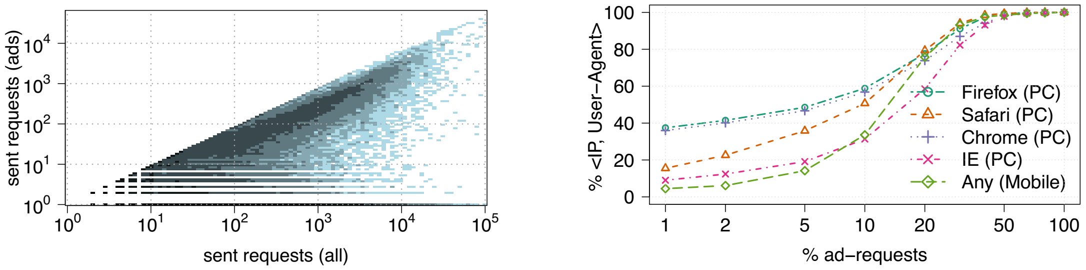
3.4.1 Annotation of active users

Our approach is to manually label the `User-Agent` strings in a subset of the data. We use this as a starting point to subsequently classify the entire data set. Our starting subset is the active users. More precisely, we select tuples that issue more than 1K requests (the heavy hitters) e.g., those corresponding

[1] We here use term "user" to refer to the pair formed by an `IP` address and a browser's `User-Agent` string.

[2] We use term "ad", "ad object" and "ad request" interchangeably in this work to refer to a request that is blacklisted by *EasyList* and its derivatives, as well as those that are blacklisted by *EasyPrivacy*. We also use this term to refer to requests that are whitelisted by the list of *non-intrusive advertisements*.

[3] Note that fetching and displaying an advert can involve several executions and thereby multiple requests [79].



(a) `RBN-2` heat map: Number of total requests (x-axis, log-scale) vs. ad-requests (y-axis, log-scale) per IP, User-Agent pair. Darker points indicate more pairs. Most pairs issue a significant number of ad requests. Other pairs issue many requests but just a few ad requests.

(b) `RBN-2` ECDF: Percentage of ad-requests (x-axis, log-scale) per active browser (those sending more than 1K requests). Many of the Firefox and Safari instances have a low ratio of ad requests.

Figure 3.3: Detection of ad-blockers in passive traces

to a few page retrievals. As a result, we obtain a more tractable set of 15.2K pairs, with 1.6K unique `User-Agent` strings. Among this set of strings, we are able to manually annotate 601, which appear in 9.6K of the heavy-hitter tuples.

With this set of annotated strings, we proceed to classify browsers in the entire set of pairs. We identify 44.1K additional browsers. All these browsers generate 57.2% of the observed requests and 82.2% of the ad requests in the `RBN-2` trace. As expected, the heavy hitters issue most of these requests, i.e., 50.6% requests and 72.5% ad requests. We thus continue our study with the set of heavy hitters, i.e., the most active browsers.

We separate these browsers into *i)* mobile and *ii)* desktop versions. We identify 1.9K browsers in the mobile category. They correspond to iPhone and Android phones. These browsers issue 5.9% of both the ad and the total number of requests in the `RBN-2` trace. The desktop category accounts for the rest of requests and it is constituted by 7.7K browsers which we further separate into Firefox (3,423), Chrome (2,267), Internet Explorer (654) and Safari (1,324) browsers.

3.4.2 Inferring ad-blocker usage

With this set of annotated browsers, we continue our analysis of ad-blocker usage at our vantage point. We use two indicators: (a) ratio of ad requests, which applies to any ad-blocker, and (b) HTTPS connections to check for filter updates, which is specific to *Adblock Plus*.

First indicator: low ratio of ad requests. We use this indicator to detect the presence of an ad-blocker from the HTTP requests. Our motivation for this indicator stems from the insights of the active measurement study (see §3.2.3). The rationale is that users that installed an ad-blocker retrieve significantly smaller number of ads than those that have not.

Figure 3.3(b) shows the empirical cumulative distribution function (ECDF) of the percentage of ad requests per browser for the annotated set of active browsers. Since not all filter lists are installed by default, e.g., the *EasyPrivacy* list, we only consider ads classified by *EasyList*, which indeed is installed by default. In this context, 40% of the Firefox and Chrome active browsers issue less than 1% ad-requests; they qualify as ad-blocker candidates. By contrast, only for 18% of the Safari and 8% of the IE instances the ratio of ad requests is below the threshold. This can be due to the fact that installing and ad-blocker like *Adblock Plus* in these browsers is a bit more cumbersome and thus might deter their usage. Based on these observations and the results from the active measurement study (see §3.2.3), we set the threshold for identifying ad-blocking browsers to 5%, to *i)* tolerate mis-classifications due to content type, and *ii)* take into consideration users that disable blocking for specific sites.

Second indicator: downloads of filter lists. As mentioned above, *Adblock Plus* frequently checks for updates to the filter lists. These updates typically occur during browser bootstrap or when a list soft-expires (see §3.2.2). Thus, we can estimate the number of *Adblock Plus* users by monitoring *EasyList* downloads. However, the *Adblock Plus* browser extension uses HTTPS to download the lists. Hence, we cannot differentiate between multiple browsers hidden behind a single IP (e.g., in the presence of a NAT), since the `User-Agent` is not visible. Instead, we can only report the number of households in which there is at least one device with an *Adblock Plus* installation.

We find several thousands of HTTPS connections to *Adblock Plus* servers in the `RBN-2` trace. These connections are issued from 19.7% of the households in this data set. This number is slightly larger than what has been previously reported. Metwalley et al. [221] reports that the fraction of households with at least one device using an *Adblock Plus* plugin is between 10% and 18%. As stated previously, this information is not sufficient to discern browsers that likely run an ad-blocker, but we can leverage it as indicator and correlate it with the ad-ratio indicator.

Correlation of indicators to assess *Adblock Plus* usage. We proceed to correlate the ad-ratio with the *EasyList* downloads indicator. We obtain four classes as the cross product corresponding to a combination of the two indicator values. We summarize these 4 classes along with their corresponding traffic statistics in Table 3.3. We find that 46.8% of the 9.6K active browsers neither classify as an ad-blocker candidate nor contact *Adblock Plus* servers (denoted in the table with the type name A). The orthogonal case, i.e., those annotated with type C, constitute 22.2% of the active browsers population. As expected, this class is dominated by Firefox and Chrome browser `User-Agents`. These two types of browsers represent 51% and 32% of the occurrences respectively. Safari on the other hand accounts for 11%. In practice 31% of the Firefox and Chrome instances fall into this class and thereby they probably have *Adblock Plus* installed. This share is slightly larger than the official statistics [21, 27]. This difference could be explained by a permissive threshold value or by a bias due to the population at our vantage point.

There are many browsers for which the download information and the ratio of ad requests produce inconsistent outcomes (types B and D). Type-D browsers exhibit ad-blocker-like behavior, albeit they did not attempt to download *EasyList*. Browsers in this category sum up to 15.3% of the active browser population. There are two possible explanations for this apparent inconsistency. The end users might *i)* have installed a different plugin, e.g., *Ghostery* or *NoScript*, or *ii)* have requested content from sites with few advertisements. Based on the observation that other ad-blockers are much less prevalent than *Adblock Plus* [221], we speculate that the most likely cause for this inconsistency is the latter scenario.

The second type of inconsistency relates to type-B browsers. They add up to 15.7% of the active browser population. For these instances we observe an *EasyList* download but the ratio of ad requests is higher than 5%. The most plausible explanation for this contradictory information is that there may be many users in the same household, some of them using *Adblock Plus* and others not.

3.4.3 Adblock Plus configurations

There are various filter lists with different objectives available for *Adblock Plus* users. Most notably, there are lists to *i)* block and hide adverts (e.g., *EasyList*), *ii)* whitelist adverts (list of *non-intrusive*

Type	Ratio	EasyList	Instances	% requests	% ad reqs.
A	X	X	46.8%	22.5%	46.3%
B	X	✓	15.7%	8.1%	15.8%
C	✓	✓	22.2%	12.9%	6.5%
D	✓	X	15.3%	7.1%	4.0%
Any	-	-	9.6K	50.6%	72.5%

Table 3.3: Ad-blocker usage: classification and statistics for the annotated set of active browsers using the indicators *i) Ratio:* low ratio of ad requests, i.e., $\leq$5%, and *ii) EasyList*: HTTPS connections to an *Adblock Plus* server.

ads), and *iii)* protect user privacy by blocking Web trackers (e.g., *EasyPrivacy*). To further elaborate on ad-blocker usage we dissect our corpus of adverts by the list that triggered the classification.

Our first observation is an unexpected high share of ad requests among the likely *Adblock Plus* users (type-C). For this observation, we refer to Table 3.3, which reports the contribution of each user category to the total number of ad requests in `RBN-2`. One would expect the relative share of ad requests for *Adblock Plus* users to be close to 0%. Instead, we find a share of 6.5% ad requests. The explanation for this high percentage is that our threshold-based classification is based on *EasyList* hits. We use this list because it is by default activated upon *Adblock Plus* installation; along with the list of *non-intrusive* advertisements whitelist. However, the numbers that we report in Table 3.3 correspond to all hits, including those triggered by the other lists, i.e., *EasyList* derivatives and *EasyPrivacy*. Indeed, we observe that 82.3% and 11.1% of the positive classifications for *Adblock Plus* users relate to filters in *EasyPrivacy* and in the list of *non-intrusive ads* respectively. This observation motivates the consequent analysis about the lists that *Adblock Plus* users subscribe to.

EasyPrivacy. Metwalley et al. [221] report that 77% of the users contact a tracker immediately after they start browsing the Web. We can leverage this observation to estimate the extent to which (if at all) an *Adblock Plus* user interacts with a tracker. Our assumption is that such interactions should only occur for those users who do not install the *EasyPrivacy* list. We observe that only 0.1% of the *non-adblock* users do not issue requests matching *EasyPrivacy* filter rules, i.e., almost every user contacts a tracker. By contrast, the corresponding fraction for *Adblock Plus* users is 5.1%. If we use a more permissive value, i.e., 10 requests to account for mis-classifications, then the percentage of *Adblock Plus* users that likely installed *EasyPrivacy* is 13.1%. Overall, we see a consistent difference around 15% using different values.

In light of these observations, we speculate that most *Adblock Plus* users, i.e., more than 85%, do not subscribe to *EasyPrivacy* but just to *EasyList*. In fact, this argument is supported by a blog post from the *EasyList* maintainers, which reported back in 2011 that only 4.1% of their 12 million users subscribed to *EasyPrivacy* [14]. The key take-away is that it seems that *Adblock Plus* users install this software to block annoying advertisements but do not configure it to protect their privacy.

Non-intrusive adverts. We conduct a similar analysis to elaborate on the prevalence of the *non-intrusive ads* list. We find that *Adblock Plus* users issue 7.9% of the total number of whitelisted requests. In comparison, *non-adblocker* users generate 37.9% of this type of requests, despite the fact that the population of this set is almost twice the population of the former set. This first observation highlights the importance of this list for advertisers and publishers alike, i.e., *Adblock Plus* users still generate a significant fraction of ad requests.

When trying to elucidate whether an *Adblock Plus* user opts out of this whitelist, we find that 11.8% of the *Adblock Plus* users issue no requests of this kind. The corresponding percentage for *non-adblock* users is 6.1%. The reason why many *non-adblock* users do not issue ad requests of this kind is because this type of adverts are less prevalent. At less than 10 issued whitelisted requests, the difference between both groups is roughly 20%, a pattern that is repeated across different values. This is, perhaps, an indication that at most 20% of the users actually deactivate this list, thus disabling the whitelisting of adverts that conform the non-intrusive ads guidelines. We emphasize a word of caution on this last statement and remark that corroborating this observation would require to conduct an analysis of the browsing patterns of *Adblock Plus* users.

Summary. Our main observation is that a significant fraction of the most active users in our traces browse the Web with *Adblock Plus*, i.e., 22.2% of the active users. This extension is especially popular among Chrome and Firefox users, i.e., 30%; and less popular among Safari and Internet Explorer users. We find that most *Adblock Plus* users do not install the *EasyPrivacy* list that protects them from trackers. Hence, it seems that *Adblock Plus* users are mostly interested in blocking ads rather than protecting their privacy. However, this might be an awareness problem. We also find that most *Adblock Plus* users probably do not opt out from the list of acceptable ads (*non-intrusive ads*) which is enabled by default. In fact, we find that the set of heavy-hitter *Adblock Plus* users still generates a substantial number of such ads, even when compared to *non-adblock* users. This observation suggests that conforming to the acceptable ads guidelines may benefit some players in the advertising domain.

3.5 Characterization of ad-related traffic

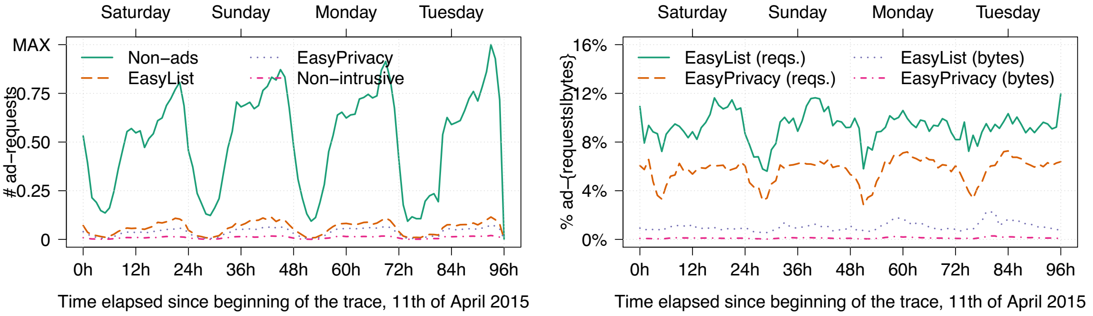
One under-explored aspect of the ad-scape is the prevalence of ads that end users experience during *regular browsing sessions*. Most previous work (see e.g., [79, 199]) has relied on active measurements and therefore cannot capture how the average user interacts with the *ad-scape* while browsing the Web. Thus, in this section we use our passive measurement methodology to study basic properties of ad traffic in the wild. Concretely, we investigate *i)* basic ad-traffic properties, *ii)* content-related properties of the observed ads, and *iii)* whitelisted ads.

3.5.1 Traffic patterns

To comment on the temporal characteristics of the ad-traffic we conduct our analysis on the `RBN-1` trace, as it spans a longer period than `RBN-2`. We find that 17.25% and 1.13% of the requests and bytes respectively correspond to ad objects in the `RBN-1` trace.

We highlight the variability of ad traffic in Figure 3.4(a), where we depict a time series for the number of requested Web objects vs. the number of ad requests using bins of 1 hour. The non-ad requests manifest the characteristic time of day and of week pattern of residential networks. During the night there are relatively fewer requests and the busy time is in the evenings, right before midnight. On the weekend there are fewer requests than during the week, in particular on Saturday. Moreover, the lunch break is also clearly visible. Surprisingly, the ad-related requests do not show the same pattern. To visualize these differences we plot in Figure 3.4(b) the percentages of ad requests and ad bytes over time. Here we only consider ads reported by filters in *EasyList* and *EasyPrivacy* (excluding *Non-intrusive ads*). The figure reveals that, surprisingly, the ratio of ad requests also manifests a diurnal pattern, ranging from 6% up to 12%, instead of having a constant rate.

To explain this surprising diurnal pattern we offer two possible explanations. The first one is that users request different content and that the pages that serve this content have a different ratio of advertisements. For example, streaming video chunks might result in a very low ratio of ads. Another example



(a) # of ad requests vs. non-ad requests

(b) % of ad requests and ad bytes according to *EasyList* and *EasyPrivacy*

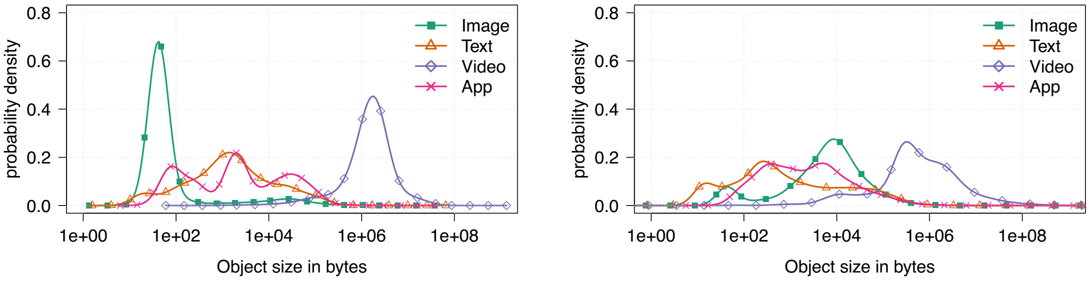
Figure 3.4: Time series for ad and non-ad traffic (1h time bins, `RBN-1` trace). Ad traffic exhibits a daily pattern, both in terms of number of requests and in terms of its ratio to all requests in the trace.

Content-type	**Ads**		**Non-Ads**	
	Reqs	Bytes	Reqs	Bytes
`image/gif`	35.1%	14.1%	3.5%	0.7%
`text/plain`	28.7%	34.2%	14.3%	2.1%
`text/html`	14.4%	11.8%	7.6%	1.1%
`-`	11.8%	5.4%	28.7%	63.4%
`application/xml`	4.5%	2.2%	1.8%	0.2%
`image/png`	1.9%	1.6%	5.1%	0.8%
`image/jpeg`	1.8%	5.2%	19.8%	3.4%
`application/x-shockwave-flash`	1.4%	8.1%	0.2%	0.1%
`video/mp4`	0.0%	10.9%	0.3%	8.6%
`video/x-flv`	0.0%	5.4%	0.1%	3.1%

Table 3.4: Trace `RBN-1`: ad traffic by *Content-type*.

are Web site categories, e.g., the *news* category has more objects than other categories (see [102] for a detailed study about the complexity of popular Web sites). Our second explanation for the diurnal pattern is that the share of ad-blockers users varies at different times of the day. We leverage the classification described in §3.2.3 to investigate this for `RBN-2`. Our finding is that at peak time the number of *non-adblocker* active users is twice the number of active *Adblock Plus* users. By contrast, during the off hours the number of active *Adblock Plus* and *non-adblocker* users is roughly the same.

Our next question relates to the type of adverts that we see. To answer it we dissect the ad requests by the list that triggers the classification. *EasyList* causes significantly more hits than *EasyPrivacy*, which in turn triggers more classifications than the *Non-intrusive ads* list. More precisely, *EasyList* classifies 55.9% of the ad requests in `RBN-1`. The *EasyPrivacy* is responsible for 35.1% of the requests. The *Non-intrusive ads* list triggers the remaining matches. We observe the same trend in `RBN-2`.



(a) Ad objects

(b) Non-Ad objects

Figure 3.5: PDF of the object size distribution of the requests according to their MIME type (`RBN-1`). Ad-related objects exhibit characteristic sizes.

3.5.2 Web objects

Next, we consider the types of the requested ads (e.g., image vs. video ads) and their prevalence. Therefore, we analyze the `Content-Type` of the Web objects in `RBN-1`. Table 3.4 shows the most prevalent objects according to the MIME type reported by *Bro* HTTP analyzer along with their corresponding contribution to the total traffic and to the ad traffic in terms of requests and bytes. Most ads are either *image/gif*, *text/html*, or *text/plain*. The fraction of bytes is dominated by *text/html* objects, while the fraction of bytes for the type *image/gif* is relatively small. The latter is not surprising given that many of the ad objects used to track users are small, i.e., 43 bytes. At the other extreme are videos (i.e., *video/mp4, video/x-flv*), flash objects (i.e., *x-shockwave*), and non-*gif* images such as *image/jpg*. All these types contribute a higher fraction of bytes than requests.

To highlight the different distributions, we show in Figure 3.5(a) the density of the ad-object size on a log scale separated by the `Content-Type`. We consider four different classes: images (*gif*, *jpeg*, and *png*), text (*html*, *plain*), video (*mp4*, *flv*), and applications (*xml*, *flash*). The density of the logarithm highlights that most images are very small (43 bytes), while most videos are rather large ($>$ 1 MByte). If we compare these values to the typical object size of non-ad objects, see Figure 3.5(b), we notice significant differences. Most non-ad videos are smaller than ad videos, while most non-ad images are larger. One of the reasons is that most video-streaming providers split regular videos into multiple chunks and each chunk corresponds to one Web object. Since most video ads only last for 15-45 seconds and advertisers expect end users to see the complete video stream, chunking may be considered to be unnecessary. Moreover, most ad videos typically have a length in the same order of magnitude. Surprisingly, we see that non-ad text objects are likely to be smaller. These are likely requests involving high-interactive sites, e.g., for auto-completion or for suggestions.

3.5.3 Non-intrusive advertisements

Next, we look at the relevance of the list of *non-intrusive ads*. On the one hand, ad-blockers threaten the financial backbone of the Web. On the other hand, they also ensure some balance by preventing ads from becoming too intrusive [33]. The result is the *Non-intrusive ads* list which whitelists adverts and is enabled by default. Indeed, according to Financial Times [7] some companies including Google, Microsoft and Amazon, pay money to *Adblock Plus* to be whitelisted. But what is the effect of this list? Given our vantage point and our ability to classify traffic the same way *Adblock Plus* does, we can investigate the impact of the whitelist using the `RBN-2` trace.

We first ask how many of the ad-related requests match the whitelist. This is the case for 9.2% of the ad requests. While this number may seem low at first, we calculated it using also the hits triggered by an *EasyPrivacy* filter. If we restrict ourselves to ad requests identified by *EasyList* and *Non-intrusive ads*, then the percentage that is subject to whitelisting grows up to 15.3% viz., these adverts would not be blocked by the default *Adblock Plus* installation.

List accuracy. However, these numbers likely *overestimate* the real impact that this list has. We manually inspected the filter rules and found some anomalies: some rules are overly general, e.g., they whitelist an entire domain rather than specifically addressing ad-related parts. For example, many requests match the `@@||gstatic.com^$document` filter rule, which whitelists the entire `gstatic.com` domain. This domain hosts unsuspected fully qualified domain names (FQDNs) e.g., `fonts.gstatic.com` and services such as *Street View*. Hence, this filter list may also whitelist non-ad-related traffic. On the other hand, and referring to the previous example, font objects may very well be necessary to display an advertisement e.g., *Google's AdSense*.

Therefore, we ask how much of the whitelisted traffic would have been otherwise blocked by a blacklist, i.e., when *Adblock Plus* users choose to stop allowing *non-intrusive* advertisements. The ratio is surprisingly small: only 57.3% of the whitelisted requests would have been blacklisted. Moreover, 23.2% of those would be filtered by *EasyPrivacy*. These observations highlight that the list of *non-intrusive ads* should be handled with caution when trying to identify adverts. In the remainder of this subsection we consider only those whitelisted requests which "match the blacklist".

Publishers. Recall, we can identify the main page that originated the request using the methodology described in [305]. We find 991 unique FQDNs with more than 1K blacklisted requests to which we can associate 84.0% of the total requests blacklisted by *EasyList* and its language derivatives. The *non-intrusive ads* list whitelists 8.6% of these requests. We find that some sites in the *dating*, *shopping*, *translation*, *audio and video streaming* categories, as well as some sites in the *mixed content* categories benefit the most from the whitelist.[4] On the other side, we find that most of the sites without whitelisted requests belong to the *adult content* category, followed by Webs in the *mixed content* category. Here, we also find another one in the *file sharing / video streaming* category. It is not surprising to see that these sites are not whitelisted. However, it is surprising to find a few instances of popular sites in the *news* category (in fact they appear in the Alexa top 1K). Thus, an *Adblock Plus* user would block every ad-related request related to these sites. We manually checked these sites with *Adblock Plus* to corroborate this observation.

Ad-tech companies. We repeat the same analysis to shed light on how the *non-intrusive ads* list affects ad-tech companies. We select FQDNs with more than 10K blacklisted requests. These domains sum up to 82.1% of the total blacklisted requests, from which 11.1% are whitelisted. Like in the publisher case, we see a mix. The list of *non-intrusive ads* whitelists 47.9% of Google's requests (recall we exclude HTTPS traffic). Some of its services do get most of their requests through (e.g., analytics, ad-services) while others do not. While the bulk of these requests corresponds to Google, we also see other companies benefiting from the *non-intrusive ads* list. One particular example is a *technology/Internet* Web site that operates its own ad-platform, for which the *non-intrusive ads* list whitelists 94% of the otherwise blacklisted requests.

Summary. Our main observation is that a significant share of the requests and bytes in a residential broadband network are due to online advertising e.g., 17.25% and 1.13% of the requests and bytes respectively for trace `RBN-1`. We observe that ad-related traffic exhibits a different diurnal pattern than regular traffic. Moreover, ad objects have more characteristic sizes than non-ad objects. In ads *gif*

[4] We use `http://sitereview.bluecoat.com` to classify Web sites into categories.

images dominate in terms of number of requests, followed by *text/plain* objects, which in turn dominate the share of ad traffic in terms of bytes.

Moreover, our analysis of the potential benefits of whitelisting i.e., the list of *Non-intrusive ads*, shows that some content publishers substantially benefit from this list, while others not. Among the latter are sites in the *adult* category. However, we also find popular sites in the *news* category among them. Some ad-tech companies also benefit from this list; for instance Google, which carries the bulk of the whitelisted traffic and for which 47.9% of the ad-related traffic is whitelisted. Another example is a popular *technology/Internet* Web site. This site operates its own ad-platform, for which the *non-intrusive ads* list whitelists 94% of the otherwise blacklisted ad traffic.

3.5.4 Server-side infrastructure

Next, we turn our attention to the infrastructure that serves ad-related objects. This study is motivated by the need to better understand the ad-scape [12], a complex and diverse ecosystem composed by hundreds of companies that provide numerous services and closely interact with each other, e.g., ad-networks and exchanges. In this section we study the server-side infrastructure from the perspective of an end user using *Adblock Plus*. Namely, which infrastructures end users contact, how often and how many.

The first aspect of the *ad-scape* infrastructure are the characteristics of the Web servers that deliver ad objects to the end users. We use the term server to refer to an IP address. Note, that any of these addresses may on the one hand be only a front-end of a large server farm or on the other hand a server that is co-located with other virtual servers. We find 29.0K and 19.6K servers in `RBN-1` that serve ad objects according to *EasyList* and respectively *EasyPrivacy*. Some servers (i.e., 5.2K) serve objects matching both lists. As one may expect—like almost every other distribution in the Internet—the distribution of requests per server is heavy-tailed (not shown). If we use only *EasyList* driven classification, the median number of ad objects per server is 7, the mean is 438, and the 90^{th} / 95^{th} / 99^{th} percentiles are respectively 320 / 1.1K / 6.8K ad objects. The busiest server in the `RBN-1` trace, which is operated by Liverail, received 312.3K ad requests in total.

The second aspect relates to the objects served by these servers: do they exclusively serve ad-related objects or do they serve regular content as well? One argument for the first case is that by now there is a separate infrastructure and market dedicated to the ad-tech ecosystem. The opposing argument is that one can take advantage of synergy effects by delivering ads via the same infrastructure as regular content. We find 222.2K servers in `RBN-1`. For 21.1% of them we classify at least one request as an ad object. These IPs serve 54.3% of the total number of non-ad objects in `RBN-1`. However, about 6.9K servers deliver exclusively ad objects. Here, we consider that a server is exclusively dedicated to deliver ad objects if our methodology identifies more than 90% of its requests as adverts. We consider this reasonable since our methodology may not be able to identify all ads and thus provides only a lower bound. We use the previous threshold to find 10.1K ad servers, which altogether deliver 32.7% of the adverts. Likewise, we define the notion of “tracking server” to refer to the set of servers that only serve ad-related objects identified using *EasyPrivacy*. We find 3.3K of these servers, which deliver 18.8% of all the ad-related objects reported by *EasyPrivacy*.

Next, we consider the Autonomous Systems (ASes) which host the ad servers since we want to understand if ad traffic is highly concentrated in a few large infrastructures. To this end we use the global routing information in order to determine the AS that is responsible for the IPs of the servers. The top-10 ASes contribute to the majority of the ad objects in the `RBN-1` trace, namely 56.8%. Among these ASes, see Table 3.5, we find four categories of players: search engines, cloud providers, CDNs and two ad-tech ASes, i.e., AppNexus and Criteo. Google leads this ranking with 21.0% and 33.9% of all ad requests

Autonomous System (AS)	**%ads relative to ad objects in trace**		**% ads relative to all objects per AS**	
	Requests	Bytes	Requests	Bytes
Google	21.0%	33.9%	50.7%	15.9%
Amazon EC2	7.0%	4.6%	19.8%	2.8%
Akamai	6.5%	19.0%	6.4%	1.0%
Amazon AWS	5.5%	1.1%	45.9%	16.6%
Hetzner	3.4%	1.4%	23.4%	3.5%
AppNexus	3.1%	0.4%	32.9%	50.2%
MyLoc	2.9%	3.0%	64.0%	14.9%
SoftLayer	2.8%	0.4%	48.5%	1.9%
AOL	2.7%	0.3%	74.7%	25.4%
Criteo	1.9%	1.1%	78.1%	88.2%

Table 3.5: Trace `RBN-1`: Ad traffic by AS for top-10 Autonomous Systems.

and bytes, respectively. The relative ratio of ad objects for Google traffic is 50.7% and 15.9% of the total ad requests and bytes to this AS. This ratio may at first seem large and one may ask if our methodology is sound. However, recall that Google switched many of its services to HTTPS—apparently mostly for their "core" content, i.e., search results and video streaming.

Among the other top contributors are also cloud providers, including Amazon, Hetzner, and MyLoc. Finding cloud providers in the list of top contributors highlights that clouds are also used by the advertisement industry. While some companies opt to manage their own AS to better control and operate their ad infrastructure, others opt to take advantage of the massive amount of resources and flexibility that cloud providers offer. Finally, CDNs like Akamai (which deploys caches within ISPs or within their own address space) and SoftLayer form the third category of ASes that serve ad objects. The presence of CDNs in this list of top contributors supports the argument that the same infrastructure that serves regular content delivers also ad objects.

The last category of ASes in Table 3.5 includes the relatively unknown ASes AppNexus and Criteo. These are companies whose main business is online advertising. They operate just a few servers (those visible in our traces), e.g., 39 for Criteo. Excluding the landing pages of these companies, we expect these ASes to mostly serve adverts, which is not strictly the case. While we classify 50.2% of the AppNexus bytes as advertisement traffic, the ratio of the requests is much lower (32.9%). Hence, our methodology probably underestimates the number of ad requests for this AS. One possible explanation for this mismatch is that the lists are conservative and do not address all possible URLs to prevent blacklisting desired traffic. Another possible explanation is that fetching an advert can require several JavaScript executions (see [79]) and thereby involve multiple non-ad requests as well. If we do not classify all requests in this chain as ad objects, except for one, the ratio of adverts per AS may decrease. In contrast to AppNexus, the percentage of ad traffic for Criteo in terms of requests and bytes is as high as 78.1% and 88.2%, respectively.

3.6 Front-end and back-end servers communication: impact on front-office Web traffic

Central to this dissertation is the distinction between front- and back-office Web traffic (§1.1). One fundamental question regarding these two categories of Web traffic relates to how they interact with each other. In this section, we tackle this particular question and investigate whether back-office activity affects the traffic dynamics of front-office Web traffic. In particular, we study the following cases:

1. **Content delivery: front-end servers fetch remote content.** Improving the end-user experience can lead to a significant increase in revenues and drive up user engagement [131, 201]. These benefits have catalyzed the competition among service companies to offer faster access to content and thereby improve the end-user experience. Two "straightforward" ways of improving the end-user experience that can be implemented by ISPs are to (a) upgrade access networks, and to (b) improve the Internet's middle mile (backbones, peering points, and transit points); both approaches, however, are expensive. Content and application providers, on the other hand, can provide better service by installing servers deep inside ISPs or through increased peering at IXPs and co-location facilities. These approaches have been reported in [111, 145, 235] and have led to an increase in back-office Web traffic.

2. **Advertisements: real-time bidding (RTB).** This term is used to describe the process of *selling* advertisement space on a per-impression basis, i.e., per user. RTB is a standard process in today's Internet (see [72, 79, 276]). The process works as follows. When a user requests an advert from an ad exchange, this exchange contacts multiple advertisers and the highest bidder among them wins the right to display the ad to the user. Advertisers can use many pieces of information to decide on their bid, including but not limited to geographical location, age, or gender. Usually, exchanges wait for around 100 ms before closing the auction [22]. Earlier work revealed the complexity of the ad ecosystem which involves multiple players [72, 248, 307].

3.6.1 Deflation of RTTs

Content providers deploy front-end servers close to end users or in strategic locations to reduce Round-Trip Times (RTTs) and thereby improve QoE. Given our vantage point, we can study if RTTs have indeed reduced over time. There exist tools like TCPtrace [236] that can extract various RTT-related statistics from TCP flows. However, as they analyze the TCP behavior, they are computationally expensive. Jiang and Dovrolis [183] proposed a lighter approach to estimate the RTT using the packets in the handshake of a TCP flow, viz. the timestamps of the `SYN`, `SYN/ACK`, and `ACK` packets. Given that our vantage point is located between servers and clients, we need to use a variation of this method. In particular, we can extract two RTTs (see Figure 3.6):

1. **Backbone latency:** this statistic corresponds to the RTT from the vantage point to the server, and corresponds to the difference between the `SYN-ACK` that the server sends and the `SYN` packet sent by the client.
2. **Access latency:** this statistic corresponds to the RTT from the vantage point to the client. To produce this statistic we have to compute the difference between the packet that acknowledges the `SYN-ACK` packet and the `SYN-ACK` packet itself.

The analysis is based on anonymized packet-level traces captured in a residential network within a large European ISP in 2008 and in January 2014. For more details on data capturing, processing and

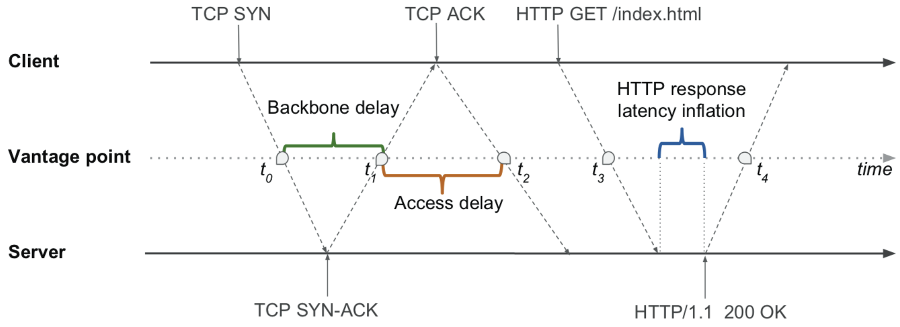


Figure 3.6: Estimation of latencies.

anonymization, we refer to [216]. We note that we do not consider flows with packet retransmissions during the TCP hand-shake as they can bias the metrics.

Figure 3.7(a) shows the backbone delay between the aggregation point of the ISP and the front-end servers that end users of the *same* residential network experienced in two different years — 2008, and 2014.[5] While the access technology, and the corresponding delay between end users and the access aggregation point, have not changed over the last six years (not shown), the delay between the aggregation point and servers has seen a significant shift towards shorter delays. A significant fraction of the requests are now terminated by servers within the same city (see the new bell curve around 1 ms delay which was not present in 2008).

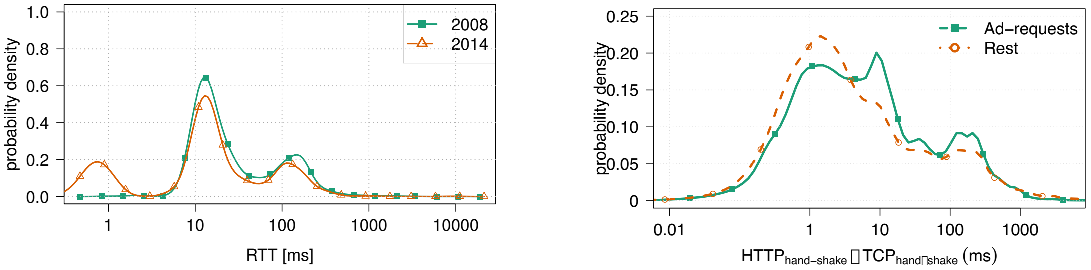
3.6.2 Inflation of HTTP-response times

Having learned that in this vantage point many requests are being served by front-end servers that are very close to end-users, we next proceed to investigate whether these servers engage in communications with back-end servers. We can use a similar approach and instead of using the TCP handshakes, we use the "*HTTP handshake*". This metric has been already proposed in related work and applied in a different context [170]. The key observation is that CDNs and ad exchanges add extra delay to the *HTTP handshake*, i.e., the time difference between the first packet in an HTTP response and the first packet of the HTTP request. We illustrate this metric on the right part of Figure 3.6 (*HTTP response latency inflation*).

Given the position of our vantage point, we cannot just use the difference between the two HTTP packets, as they may include significant network delays. To circumvent this problem we use the *backbone latency* as a proxy for the network round trip time (RTT) to the server. By doing so, we also remove biases due to the servers' locations, viz. otherwise servers in Europe would serve data faster than servers in the US or Asia. If the HTTP object is in a persistent TCP connection we still use the TCP handshake time from this connection as the delays usually do not vary that significantly within a few seconds—the expected durations of these persistent connections.

Next, we proceed to identify real-time bidding by leveraging a threshold of 100 ms for answering a request. Figure 3.7(b) shows the *density of the logarithm* of the difference between the HTTP and TCP handshake times for the RBN-2 trace, dissected into ad and non-ad HTTP transactions. The first observation is that most of the handshake time differences are small, i.e., 1 ms. They relate to noise on the network path or to the processing overhead at the server to determine the HTTP response. The

[5] We note that not necessarily all traffic is exchanged with front-end servers.



(a) Delay between access aggregation point and front-end servers in 2008 vs. 2014.

(b) Real time bidding in `RBN-2`: difference between the HTTP and TCP hand-shake latencies dissected by type of request. The more pronounced modes at 10 ms and 100 ms suggest that ad requests typically involve more back-end functionality than non-ad requests.

Figure 3.7: Inferring back-office activity from front-office traffic.

second observation is that while most of the non-ad objects have a short handshake time, namely less than 10 ms, ads have— more often than non-ads—larger handshake times. This can be seen from the three modes at 1 ms, 10 ms and 120 ms in Figure 3.7(b). Compared to the non-ad objects a much larger share of ad objects exhibits a time difference between handshakes of more than 100 ms. This suggests the presence of *back-offices*, which include ad-exchanges to enable RTB and CDN to fetch objects from other distant servers.

In fact, a manual inspection of the fully qualified domain names with such large handshake time differences ($\geq$90 ms) reveals that these host names belong to ad-tech companies. For example, Google's DoubleClick, which offers RTB, contributes to 14.5% of the ads in this range. We also find other organizations offering RTB, e.g., Mopub (an RTB exchange for mobile in-app ads), the Rubicon project, Pubmatic or Criteo. Each of these organizations contributes roughly 5% to the total number of ads in this range. We also find Web tracking companies, e.g., AddThis.

3.7 Summary

The goal of this chapter is to expose and characterize the interactions between the end users and the ad ecosystem. We conduct a passive measurement study using traces collected at a residential broadband network of a major European ISP with two objectives in mind. First, elaborate on the prevalence of ad-blockers to inform the ongoing debate regarding these tools. Second, complement related work by characterizing ad traffic at this vantage point using *Adblock Plus* functionality.

Our main observation is that a significant fraction (22%) of the most active users in our traces browse the Web with *Adblock Plus*. Surprisingly, we find little evidence that *Adblock Plus* users install the *EasyPrivacy* list of filters, which aims to protect end users' privacy by blocking trackers. Likewise, our results suggest that most *Adblock Plus* users do not opt out from the list of acceptable ads that is enabled by default in *Adblock Plus*. Based on these observations we conjecture that *Adblock Plus* users are mostly interested in blocking annoying ads rather than protecting their privacy, or that they are not aware of these options or how to change them.

Motivated by these observations, we also investigate the potential benefits of whitelisting, i.e., the list of *Non-intrusive ads*. This list can be tremendously beneficial to some content publishers and ad-tech companies given the number of *Adblock Plus* users. We find that 9% of the ad-related requests are whitelisted by this list, while 56% and 35% of the ad-related requests are blacklisted by *EasyList* and *EasyPrivacy*, respectively.

Overall, we find that 18% and 1% (requests and bytes) of the total traffic at our vantage point relates to ad traffic. This share of ad traffic is distributed across different infrastructures, including content and cloud providers and CDNs. However, these infrastructures are concentrated in a few Autonomous Systems. The most prevalent Autonomous Systems include the expected players, e.g., Google (which dominates the list in terms of requests and bytes). We further observe that some ad-tech companies opt to either deploy their infrastructure in clouds or rely on CDNs, while others choose to manage their own AS, e.g., AppNexus and Criteo. Moreover, our data suggests that often the same infrastructure serves ad content as well as regular content, although some servers do indeed only serve ad-related content.

Finally, we also show that ad-related requests typically involve more back-end functionality than non-ad requests. This complexity results in higher observed response times for ad objects than non-ad objects. These latency inflations suggest the presence of real-time bidding within ad exchanges as well as CDN back-office activity, which motivates the next chapter of this dissertation.

4

Back-office Web traffic: Content delivery, real-time bidding, and crawling

Having learned that a number of Web flows experience an inflation of the HTTP response times due to front-end servers communicating with back-end servers (see §3.6), in this chapter, we now focus on studying *back-office* Web traffic. Front-office traffic has long been studied, e.g., [66, 90, 103, 139, 142, 216, 243]. In contrast there is less related work on back-office traffic. There are studies which focused on such traffic in specific environments, e.g., in cellular networks [269] and within data centers [87, 88, 160]. Liang et al. [150] studied security-related aspects arising from CDN back-end communication, and Gao et al. [151] characterized DNS traffic between name servers. This chapter's focus is the study of back-office Web traffic on the public Internet that serves to realize Web content delivery, monetization, and search.

The reason why previous work has focused mainly on front-office traffic is that end-user Quality of Experience (QoE) can be analyzed by observing front-office traffic, but back-office traffic is often opaque. However, more and more Web services also depend on some back-office communication over the public Internet, e.g., to assemble Web pages, to perform search queries, to place advertisements, to conduct database transactions, to dynamically generate personalized content, etc. Thus, the back-end architecture and its performance can affect the perceived quality of a service. This dependency makes measuring and characterizing back-office traffic challenging but necessary.

Among the difficulties faced in studying back-office traffic is that it is rarely present on the network links connecting end users to the Internet. Instead, back-office traffic can generally only be observed on backbone or inter-domain links. However, existing studies of inter-domain and/or backbone traffic [152, 203, 249] have not distinguished between front- and back-office Web traffic. Indeed, the back-office traffic component for any individual link depends highly on whether the link is on any of the routes between the servers engaging in a back-office communication. Thus, observing this traffic requires a variety of vantage points. In this chapter we analyze data collected from two IXPs, multiple links of a Tier-1 ISP, and a major CDN. In such vantage points we may be able to observe back-office traffic resulting from Web content delivery, search, and advertisements. Recall, Web content delivery is responsible for a significant fraction of all Internet traffic, while advertisements (and in particular those in response to search) are responsible for a significant fraction of Internet revenues.

Given this state of affairs, we argue that back-office Web traffic is one fundamental yet largely unexplored component of the Internet's Web traffic. The contributions of this chapter are:

1. We introduce the notion of back-office Web traffic and show that its contribution ranges on average from 10% to 30% per vantage point and can even exceed 40% for some time periods. The vantage points include two major IXPs, multiple backbone links from a major ISP, and a number of server clusters from a major CDN. We explore the reasons that different levels of contributions are seen at different vantage points.
2. Our methodology allows us to identify and classify different types of back-office traffic including that generated by proxy services, Web crawlers, and ad-exchanges.
3. Our analysis demonstrates that back-office traffic characteristics differ from those of front-office traffic. Hence, it is possible to identify individual services based on their traffic patterns. We find, for example, that at one of the IXPs auctioneers have a 22% share of the back-office requests but only 1% of the bytes, while crawlers contribute respectively roughly 10% and 15% to both.
4. Our analysis of data from a major CDN confirms what we observe in the wild: CDNs deploy sophisticated back-office infrastructures. We further characterize back-office traffic for this particular CDN and show how this traffic has multiple purposes.
5. Given the volume of back-office traffic on the Internet and its importance for end-user QoE, we identify implications of our analysis on network protocols design and co-location strategies.

The rest of this chapter is organized as follows. In Section §4.1 we describe possible scenarios where an Internet's host generates back-office Web traffic. Section §4.2 enumerates the vantage points and datasets used in this chapter. We present our methodology for identifying back-office Web traffic in Section §4.3, which we use in Section §4.4 to characterize various aspects of such traffic across multiple vantage points. We present a in-depth study of back-office Web traffic at a CDN in Section §4.5. Finally, we conclude this chapter in Section §4.6.

4.1 Background

In this section we provide a brief overview of the typical (i.e., expected) communication patterns of Web services that create back-office traffic. Hereby, we distinguish four different cases: (a) proxies/intermediaries, (b) CDN services, (c) auctioneers, and (d) crawlers. Figure 4.1 provides an illustration of the expected exchange of HTTP messages. Note, however, that our analysis (Section §4.4.5) unveils richer and more complex communication patterns than those shown in the figure.

1. **Proxies/Intermediaries:** An intermediary is a network entity that acts as both a client and a server. As shown in Figure 4.1(a), a Web proxy is an intermediary that acts as a client with the main purpose of forwarding HTTP(S) requests. Thus, Web proxies send and receive requests in a temporally correlated fashion. Forward and reverse Web proxies evaluate requests, check if they can be satisfied locally, and contact a remote server only if necessary. When intermediaries act as clients, they create back-office traffic, but when intermediaries act as servers, the traffic they create can be either front- or back-office traffic. We describe how to differentiate these two cases in Section §4.3.

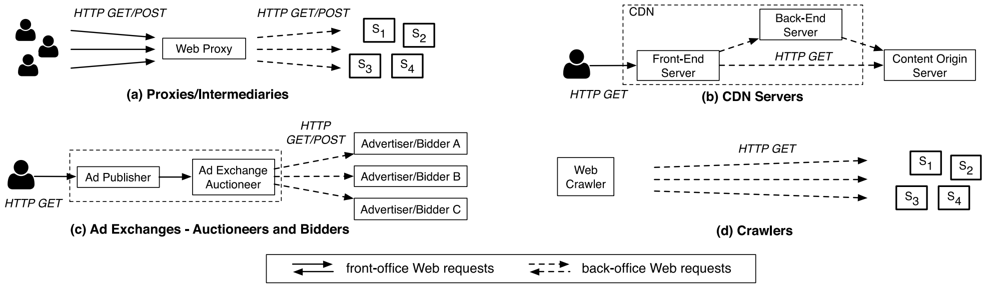


Figure 4.1: Back-office Web Traffic: typical HTTP requests made by Web proxies, CDNs, ad-exchanges, and crawlers.

2. **CDN Servers:** CDNs typically operate front-end servers (i.e., reverse proxies) close to the end user as well as back-end servers. Back-end servers either host the content in data centers or are closer to the origin content server, depending on the CDN's deployment and operation strategy (see §2.1.1). If the front-end does not have a requested object available locally, it fetches the object from another front-end, a back-end, or the origin server.
3. **Ad Exchanges – Auctioneers and Bidders:** As shown in Figure 4.1(c), advertisement exchanges consist of (i) publishers that sell advertisement space (ad space) on their Web pages, as well as (ii) advertisers that buy ad space on these Web pages. Hence, if real-time bidding (RTB) is used to place ads on a website, the visit of a Web page by an end user may trigger a large number of requests in the background. The final advertisement content is typically delivered via CDNs [72]. We note that today's Web advertisement ecosystem is complex and may involve many different types of back-office traffic, caused by a variety of different actors. In this thesis, we solely focus on RTB-related activity, i.e., back-office traffic as a result of auctioneers interacting with bidders.
4. **Crawlers:** Indexing involves requesting the Web page as well as following embedded links [99, 189]. Web crawlers typically issue an order of magnitude more Web queries than regular end users. Best practices among the major search engines ensure that crawlers have appropriate reverse DNS entries along with well-specified user agents in order to avoid being blocked by Web sites.

Hereafter, we refer to back-office Web traffic as all Web traffic that is not exchanged between end users and servers. This includes traffic exchanged between intermediaries and Web servers (e.g., traffic between a CDN front-end server and a back-end server or between a Web proxy and a Web server), as well as traffic exchanged between automated hosts such as crawlers or auctioneers and any other Web server.

4.2 Vantage points and datasets

For our study we rely on measurements collected at a diverse set of vantage points. This diverse set of traces allows us to study back-office traffic in a variety of locations, including inter-domain links, backbone links, and links connected to a major CDN's infrastructure. Table 4.1 summarizes the properties of our data sets.

1. **IXPs:** Packet-sampled traces collected at two Internet eXchange Points (IXPs), which allow us to study back-office traffic in an inter-domain environment, as exchanged between hundreds of networks [64]. The IXP traces are collected from the public switching infrastructure at two European IXPs. This includes a large IXP (L-IXP) with around 500 members and a medium-sized IXP

Type	Name	Daily traffic	Collection	Period	% TCP	% Web of TCP
Exchange	L-IXP	11,900 TB	sFlow (1/16K)	37^{th} week 2013	84.40%	78.27%
	M-IXP	1,580 TB	sFlow (1/16K)	4^{th} week 2014	97.22%	92.22%
Transit	BBone-1	40 TB	sampled (1/1K)	5-12 Feb. 2014	86.30%	64.58%
	BBone-2	70 TB	sampled (1/1K)	4^{th} week 2014	73.87%	78.74%
Content	CDN	350 TB	Server logs	24-25 Apr. 2014	95%	95%

Table 4.1: Summary of the vantage points and collected traces to study back-office Web traffic.

(M-IXP) with around 100 members. Among the member ASes there are many CDNs, Web hosting services, cloud providers, and large commercial Web sites. We collect sFlow records [267] with a 1 out of 16K sampling rate. sFlow captures the first 128 bytes of each sampled Ethernet frame, providing us access to full network- and transport-layer headers and some initial bytes of the payload, allowing for deep packet inspection (DPI).

2. **ISP:** Anonymized packet-sampled traces collected at two transatlantic links in the backbone of a large European Tier-1 ISP, providing a view of back-office traffic on long-distance links. These links carry mainly transit traffic. We collect anonymized packet traces with a random packet sampling rate of 1 out of 1K.
3. **CDN:** Web server logs from multiple servers in different locations within a large commercial CDN. These logs give us an inside view of back-office traffic created by a CDN. The logs from the large commercial CDN encompass the activity of all servers at one hosting location in each of five large cities. Each log entry contains TCP summary statistics including endpoint IPs, number of bytes transferred, and initial TCP handshake round-trip latency. In addition, we received a complete list of all IP addresses used by the CDN infrastructure.
4. **Active measurements:** Probes of IP addresses and DNS reverse lookups to identify Web servers. We also use active measurement data from the ZMap Project [134]. This data set contains a list of IPs, i.e., servers, that are responsive to `GET` requests on port 80 (HTTP) and SSL services on port 443 (HTTPS), spanning the time period from October 2013 to January 2014. In addition, we also make use of the data made public by the authors of [105, 278] that disclose the set of IPs used by the Google infrastructure. When combining Google IPs with one of our packet traces, we use the snapshot of the Google IPs that corresponds to the last day of the trace.

4.3 Methodology: Web endpoints-based classification

Given the above characteristics of back-office Web traffic (§4.1), we next describe how we identify a significant fraction of it within our data sets. Our methodology involves three steps. First, we classify all IPs based on whether they are involved in any Web activity. Second, we classify their activities as either Web client, Web server, or both—client and server. Finally, we identify auctioneers and crawlers among the clients, bidders among the servers, and Web proxies among those that are acting as both clients and servers.

We focus only on those IPs for which we see Web activity in our traces, meaning either HTTP or HTTPS activity. For this we rely on well-known signatures to detect HTTP requests (`GET`, `POST`) and responses (`HTTP/1.{0,1}`) on any TCP port. For HTTPS we use signatures to match packets to/from port 443 that contain a SSL/TLS hand-shake, i.e., `Client Hello` and `Server Hello` messages [91]. We focus on IPv4, since IPv6 traffic accounts for less than 1% of the traffic across all data sets.

Name	#IPs	Method	Client-only (%)	Server-only (%)	Dual role (%)
L-IXP	45.79M	DPI	96.90	2.74	0.36
		DPI+ZMap	93.85	2.74	3.40
M-IXP	1.9M	DPI	95.15	4.62	0.24
		DPI+ZMap	92.86	4.62	2.52
BBone-1	1.1M	DPI	92.26	7.56	0.18
		DPI+Zmap	86.62	7.56	5.82
BBone-2	4.5M	DPI	95.54	4.36	0.09
		DPI+ZMap	93.97	4.36	1.67

Table 4.2: Web activity of IPs: client-only, server-only, or both (dual) across vantage points.

The result is a set of Web server endpoints, i.e., tuples that contain the IP address and the corresponding port number, as identified using the above-mentioned signatures. We then refer to all packets that are sent to/from one of the Web endpoints as Web traffic. Our methodology ensures that in the case of server IPs also hosting other applications on different ports (e.g., email), only their Web-related traffic is considered. Table 4.1 shows the percentages of Web traffic within our different data sets. As expected, this traffic constitutes a large fraction of the TCP traffic, ranging from 64% to 95% in each data set.

Given that we have identified Web server endpoints, we next classify the Web activity of IP addresses to be client-only, server-only, or both (referred to as *dual* behavior in the following). We say that an IP address acts only as server if all of its traffic is related to its previously identified server-endpoint(s) (typically on port 80). If we see this IP address acting only as client, i.e., sending requests and receiving replies from other server-endpoints, it is classified as client only. If we see an IP address both acting as client and as server i.e., it runs a server on a specific port but also issues requests towards other servers, we classify its behavior as dual.[1]

Depending on the vantage point however, one may not see all Web activity a host is involved in. For example, a proxy server might exhibit only client-only activity when monitored in the core of the Internet, and only server-only activity when monitored in an access network. To tackle this limitation, we rely on a combination of passive and active measurements to uncover more IPs with dual-behavior as follows: we obtain a list of client IPs via DPI from our traces and then use the ZMap data set to check if these IPs respond to HTTP(S) queries. The ZMap data set provides lists of IPs that answer to a `GET` on port 80 or to an SSL handshake on port 443. Thus, if we see an IP address acting only as client, but we find it in the ZMap data set, we classify its behavior as dual.

Table 4.2 shows the classification of IPs when only relying on DPI (first row for each vantage point), as well as after taking the ZMap data set into account (second row for each vantage point). We make three observations. First, with only DPI, roughly 90% of IPs are classified as client-only across all data sets. Second, a significant fraction of the server IPs also show client behavior e.g., with DPI we see in the L-IXP trace that 11% of the total number of servers also act as clients. Third, adding more information for identifying dual behavior helps e.g., with DPI+ZMap we see in the L-IXP trace that 55% of the servers behave also as clients. Indeed, the fraction of IPs acting both as clients and servers increases significantly across all vantage points when combining active and passive measurements.

There are two main caveats when using this classification approach, which likely result in some overcounting of dual hosts on the one hand, as well as some undercounting on the other hand. One factor contributing to possibly overcounting dual hosts derives from the usage of dynamically assigned IP addresses. If a dual host is assigned different IP addresses at different times, then each of those IP

[1] Recall that requests are typically issued with *ephemeral* source port numbers.

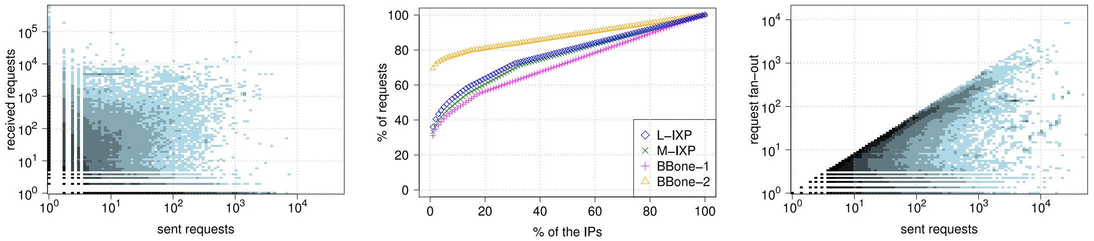
addresses may be classified as a dual host, even though at other times the same IP addresses act only as servers or, more commonly, only as clients. Dynamically assigned IP addresses are typically found in residential networks. Due to bandwidth limitations, these addresses do not serve a significant fraction of Web traffic or a significant fraction of Web requests. Nevertheless, to minimize the impact of dual hosts with dynamically assigned addresses on our statistics, we only count as servers IP addresses that appear in two consecutive snapshots of the ZMap data set, i.e., they replied to HTTP requests issued two weeks apart. On the other hand, our methodology may undercount dual hosts because is not able to detect more complex cases of dual behavior, e.g., if a server has multiple interfaces and thus sends/receives traffic from multiple IP addresses, each acting only as a client or server. Furthermore, while we can uncover the server activity of a host that acts only as a client in our traces using the ZMap data set, we lack the tools to uncover the opposite case i.e., to uncover the client activity of a host that acts only as a server in our traces.

Along with the methodology previously described, we also exploit a list of servers provided to us from a large CDN. Henceforth, we distinguish three different classifications of IP addresses, *IPs-CDN*, *IPs-DPI*, and *IPs-ZMap*, based on our confidence in their accuracy. We are most confident in the accuracy of the classifications of the CDN addresses. Based on the DPI performed on a trace, we can also be very sure that an IP at that point in time was used in the identified fashion. The ZMap data set comes with the largest degree of uncertainty, but covers a wide range of IPs. Note that the same IP address may appear in more than one of these sets of classifications.

First indicators of back-office activity. Figure 4.2(a) shows a scatter plot for all IPs seen in the L-IXP trace, where we plot each IP according to the number of sampled Web requests it sent vs. received on a log-log scale. This plot highlights the different classes of IPs. The server-only IPs only receive requests and are scattered along the y-axis. The client-only IPs are scattered along the x-axis. The dual-role IPs are scattered across both axes. While the IPs with significant activity in both roles are likely to be intermediaries/proxies, we first take a closer look at some of the other heavy-hitters, i.e., those that only act as clients.

Figure 4.2(b) shows the cumulative percentage of the number of sampled `GET` and `POST` requests issued per IP, sorted by the number of observed requests. The most striking observation from this plot is that less than 1% of all the IPs are responsible for more than 30% of all requests across the IXP and backbone vantage points. Indeed, we estimate from the sampled data that the IP that contributes most requests at the L-IXP contributes more than 5M requests per hour on average, while at BBone-2 this number is roughly 310K requests per hour. These are unlikely to be humans — rather this behavior points towards a crawler, a proxy, or an auctioneer.

Accordingly, Figure 4.2(c) shows for each IP the number of sampled requests it sent vs. the number of IPs to which these are sent, namely the *fan-out*, on a log-log scale. While some of these clients contact a huge number of servers, there are also clients that send an enormous number of requests to a rather small set of server IPs (bottom right). As shown in Table 4.3, when inspecting the types of the requests we find an unexpectedly large number of `POST` requests. These can be attributed to the intensive use of protocols for Web services (e.g., SOAP). Closer inspection shows that for clients with an extraordinary high number of requests the fraction of `POST` requests is larger when compared to clients with low numbers of requests. Based on these observations, we now differentiate between proxies, auctioneers, crawlers, and bidders.



(a) Scatterplot: # sent vs. received requests sampled per IP in the L-IXP trace.

(b) Cumulative sum: # of requests sampled per IP, ranked by activity.

(c) Scatterplot: # sampled sent requests vs. fan-out per IP in the L-IXP trace.

Figure 4.2: Web IP activity: request frequency and fan-out.

Name	Total	GETs	POSTs	CHellos
L-IXP	76.36M	71.6%	11.5%	16.9%
M-IXP	2.65M	78.9%	3.7%	17.4%
BBone-1	1.81M	88.3%	5.2%	6.5%
BBone-2	2.92M	58.1%	8.2%	33.7%

Table 4.3: Web IP activity: sampled HTTP/HTTPS requests.

4.3.1 CDNs, proxies and other intermediaries

Typical examples of dual-behavior hosts are proxies such as those deployed by some institutions and forward and reverse proxies such as those operated by CDNs and content providers (CPs). However, intermediaries can, in fact, serve many other purposes. For instance, there are many kinds of proxies in the wild that affect a significant number of connections [303] and which are not operated by CDNs or CPs. In addition, intermediaries at hosting or cloud service provider networks may not necessarily operate for the single purpose of request forwarding. While keeping this in mind, we focus on identifying some of the intermediaries that are operated by CPs or CDNs, which we hereafter refer to as *Content Delivery Proxies* (CDPs).

These are the steps we follow to classify CDPs. Along with the IPs in the *IPs-CDN* set, we select as potential candidates those intermediaries for which we sampled more than 5 requests (heavy-hitters).[2] We then check the origin AS of the corresponding subnet, and manually inspect if the WHOIS information reveals that the address is registered to a known CP or CDN. Since this check is not sufficient to reveal cases in which front-end servers and caches are embedded in ISP networks and use the address space registered to those networks, e.g., Akamai and Google Global Cache, we also check for DNS host-names and use the techniques reported in [105, 278] to attribute IPs to content providers and CDNs.

Based on the previous manual identification, we are able to classify among the list of intermediaries some of the front-ends of 8 well-known organizations such as Google, Akamai, Limelight or EdgeCast. We find more than 36K (15K) IPs in the L-IXP (M-IXP) traces. We also find CDPs that are active on the transatlantic links i.e., 9K and 19K for the BBone-1 and BBone-2 traces.

[2] Note that 5 sampled requests correspond to an estimated number of roughly 80K requests per week for the IXP traces and to 5K requests per week in the BBone links, respectively.

4.3.2 Real-time bidding: Auctioneers and bidders

The bidding process between auctioneers and bidders is generally done using Web services, such as Google AdExchange or Facebook Exchange. Bidders register with the auctioneer and provide a URI on which they accept offers from the auctioneer and reply with their corresponding bid after a maximum time threshold, often around 100ms [22].

Thus, one of the distinguishing characteristics of auctioneers is that they typically send their offers via `POST` to the potential bidders. The bidders, in turn, receive a large number of `POST` requests from a relatively small number of hosts (the auctioneers). This fits nicely with our earlier observation that there are many more `POST` requests in today's Internet than were observed in the past. In particular, an examination of traces from L-IXP over the past three years shows that the fraction of requests of type `POST` has increased by 80% over that time. Indeed, for each user request which involves an advertisement there may be multiple bidders contacted.

Given that the market for real-time bidding (RTB) is heavily concentrated among a relatively small number of players accounting for the major share of the RTB activity [307], prime candidates for auctioneers are IPs sending large numbers of requests to a comparably small set of IPs (bidders), which in turn receive a large number of requests from a relatively small number of IPs. As bidders can provide customized URIs to auctioneers, we cannot identify auctioneers and bidders in a straightforward manner using payload signatures. Instead, we identify auctioneers and bidders based on partially available URL strings as follows: we first obtain a list of partial URLs sent by the heavy-hitters and select those IPs whose HTTP requests contain in the absolute path and/or query parts of the URL strings such as `ad`, `bid`, or `rtb`. Then, for each of these IPs, we check if its corresponding subset of requests has a fixed structure, i.e., we observe many repetitions of the same partial URL (only the value fields in the query part change), and we mark these IPs as candidates for potential auctioneers. We then manually validate that these IPs are operated by organizations offering RTB services by relying on meta-information such as reverse DNS, WHOIS, and publicly available API documentation of some ad exchanges. Having a list of auctioneers, we now inspect the corresponding destination IPs, further manually verify the corresponding URL strings to be bidding-related, and mark these IPs as bidders.

With this method we are able to manually identify 316 IPs used by auctioneers and 282 IPs used by bidders in the L-IXP trace. We were not able to identify bidding activity in the M-IXP trace, and also did not identify any ad exchanges co-located at M-IXP. Nor did we find any bidding activity in the backbone traces, perhaps because of the high delay in transatlantic links.

4.3.3 Crawlers

One of the distinguishing characteristics of Web crawlers is that they issue many Web queries and then upload their results to the search infrastructure. The queries constitute a large number of `GET` requests sent to a diverse set of servers belonging to different organizations.

For each data set we use the heavy hitters, in terms of `GET` requests, as candidates. Specifically, we pre-select IPs for which we sample at least five queries and then try to verify that they are Web crawlers as follows. It is a best common practice to make crawlers clearly identifiable by setting up reverse DNS entries and including an appropriate user-agent with each request.[3] Thus, we search for host-names that include indicative strings such as *bot, crawl, spider* and select those that can be either automatically

[3] See, for example Google https://support.google.com/webmasters/answer/80553?hl=en, and for Microsoft Bing http://www.bing.com/webmaster/help/how-to-verify-bingbot-3905dc26

Name	CDPs	Bidders	Auctioneers	Crawlers	Other
L-IXP	36054	282	316	3920	151095
M-IXP	15116	0	0	541	4417
BBone-1	9330	0	0	81	1214
BBone-2	19890	0	0	894	2669

Table 4.4: Back-office: IP classification.

validated via the user-agent or validated with a manual inspection of the corresponding reverse DNS entry.

With this method, we identify 3920 and 541 crawlers in the L-IXP and M-IXP traces. Surprisingly, we also find crawlers in the backbone traces: 81 and 894 for the BBone-1 and –respectively– BBone-2 traces. We see activity from well-known search engines e.g., Google, Bing, Yandex, and Baidu, as well as from smaller search engines and research institutions. To put these numbers into perspective, we use the ZMap reverse DNS data set (i.e., all IPv4 PTR records) and search for host-names of crawlers belonging to three major search engines, which are well-defined in publicly available documents provided by these engines. The percentage of crawler IPs for which we see activity in the L-IXP trace is 7%, 23%, and 51% for three of the major search engines.

4.4 Characterization of back-office Web traffic

After identifying IP addresses that can create *back-office* Web traffic, we next discuss them in more detail. Then, we investigate the characteristics of the traffic they generate.

4.4.1 Characterization of IP addresses

We find that most IPs that are part of the Web ecosystem are clients, but there are a substantial number of Web intermediaries across the vantage points e.g., just by inspecting the L-IXP trace with DPI, we find that 11% of the Web servers also act as Web clients. However, after combining our passive data with active measurements, we discover that many of the client IPs in the traces also act as servers, which is not visible when purely relying on passive data. As a consequence, the number of IPs with dual behavior increases e.g., for the L-IXP trace more than 50% of the server IPs exhibit also client behavior. As shown in Table 4.4, after we inspect the heavy-hitter IPs, we are able to find activity from content delivery proxies, ad auctioneers, ad bidders, and crawlers in the L-IXP trace, as well as crawling and intermediary activity as seen from the other vantage points. To better understand which players are involved in back-office services, we next take a closer look at its components in the L-IXP trace.

Auctioneers and bidders. We identify more than 300 IPs that are auctioneers. These IPs are operated by four different organizations that offer real-time bidding: Two search engines, an online social network, and a major Web portal. Each of these organizations operates at least one AS and the IPs are hosted in the corresponding AS. With regards to the number of IPs per AS we see a rather uneven distribution: The top one hosts 83% of the IPs. The same holds for the distribution of the number of requests: the top organization is involved in 55% of the bids, the others in 32%, 10%, and 3%.

These auctioneers communicate with a rather small set of bidders (282 IPs). The IXP data shows that many of the auctioneers are co-located with the bidders (both the AS of the auctioneer and the AS hosting the bidder are members of the IXP), with the bidders residing in 42 different ASes. This confirms that bidders are following the recommendations by the auctioneers to locate their servers close by in order to adhere to the strict deadlines of auction-based advertisements. Bidder IPs are typically contacted from all four identified auctioneering organizations. Thus, bidders do not use different IPs for different auctioneers and often cooperate with all of them. The likely motivation is that advertisers try to maximize their bidding opportunities (i.e., receiving offers from all organizations). Moreover, at first glance the number of bidders may appear small but this is mainly due to aggregation. Indeed, most IPs belong to advertisement aggregators.

With regards to the ASes that host the bidders, we find, surprisingly, that a very large hosting service provider dominates with a share of 34%. Indeed, even the remaining bidders are mainly located in the ASes of other Web hosting providers. This finding indicates that today's major players in the Web ecosystem often do not operate their own infrastructure either in terms of an AS or in terms of a data center. They rely instead on cloud services and Web hosting companies. The second AS with the most bidders belongs to, surprisingly, a company that operates a search engine. Indeed, this search engine is involved in all services: it is a search engine, an auctioneer, and even bids on other ad-network auctions. This finding illustrates the complexity of the advertising ecosystem, where different types of business relationships exist between organizations that offer multiple services to both advertisers and publishers, and who may also partner with some of them.[4]

Crawlers: We identify more than 3K crawler IPs from 120 different ASes. Among the ASes, there are two hosting more than 72% of the crawler IPs. These are related to two popular Web search engines. We also see crawlers of 3 other well-known search engines, each with roughly 100 crawlers. Then there is a gap with regards to the number of crawlers per AS as the remaining crawler IPs are hosted in many different ASes. Inspecting the user agent and the reverse DNS entries allows us to conclude that these are mainly associated with specialized search engines.

With regards to the number of requests, the four top contributors all belong to major search engines. The top three/four account for 94/96% of the requests. The fourth accounts for only 2% of the requests. Even though the crawling activity is directed towards more than 4.2K ASes, a single AS receives more than 43% of the requests — a Web hosting provider. The second most popular AS is another hosting provider and both are members of the IXP. Overall, the members account for more than 80% of the received crawling requests. In terms of request popularity, the first AS that is not a member of this IXP is yet another hosting provider and receives roughly 1% of the requests. Overall, the top crawling search engine AS and the top receiving hosting provider AS account for more than 20% of all crawling-related requests.

Content delivery proxies: We identify more than 30K intermediary IPs from 8 well-known CPs and CDNs, scattered across hundreds of different ASes, interacting with IPs from more than 1K different ASes. The CDPs are responsible for roughly 17% of the requests from heavy-hitter IPs. While one expects many of the front ends to be located in ISP networks, a close inspection of destination IPs reveals that some of the back-end servers are also located within the ASes of ISPs, and not in the AS of the CDN. In fact, we observe requests for the content of a major online social network (OSN) where both source and destination IPs are operated by a major CDN, but neither of the endpoints is located within the AS of the CDN or OSN. We also find other more typical scenarios, such as CDNs fetching content directly from OSNs and from large-scale hosting provider ASes.

Other intermediaries: The rest of IPs in the intermediary list (roughly 151K) are located in more than 7K ASes. They contact 399K servers in 10K different ASes. While we strongly suspect that most of these are indeed Web proxies, we cannot be certain. Indeed, on the one hand, one of the heavy-hitters

[4] For an example of such a complex relationship see http://ir.yandex.com/releasedetail.cfm?releaseid=828337

IPs in this set — an oddball — is hosted in an unexpected AS. This oddball IP is serving both ad-related images to a CDN and acting as a Web auctioneer. On the other hand, we see several organizations that use resources from cloud services to set up their own virtual CDNs. A close analysis of which ASes are hosting the heavy hitters shows that most of these ASes are hosting and/or cloud service providers (8 out of 10). There is, however, more diversity in the destination ASes: we find hosting providers, CPs, OSNs and CDNs. We see that a single hosting/cloud service provider is responsible for 21% of the requests issued by IPs in this set. This observation highlights the importance of cloud service providers in the back office of the Web ecosystem once again.

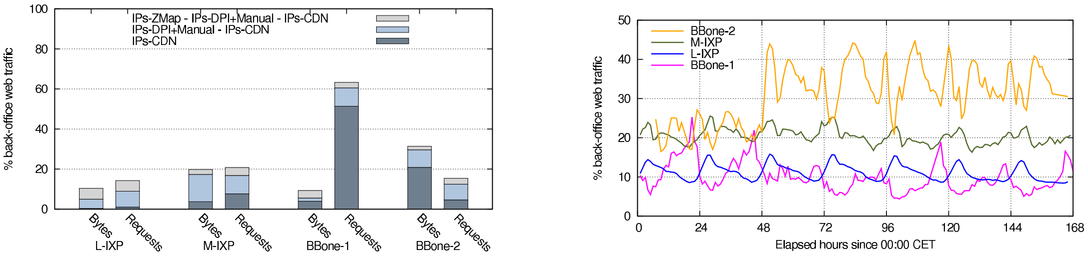
4.4.2 Quantification across vantage points

The methodology presented in Section §4.3 allows us to classify IPs as Web crawlers, auctioneers, bidders, and intermediaries such as CDPs. Thus, we can now quantify the amount of back- and front-office Web traffic. For a packet to be classified as back-office traffic, we require that both the source as well as the destination were previously identified as part of the back-office ecosystem. More specifically, we require that the source IP address was previously identified as belonging to an intermediary, crawler, or auctioneer, and the destination IP:port pair matches one of our identified Web server endpoints. We then tag this packet as back-office traffic, issued by the source, namely a crawler, auctioneer, CDP, or other intermediary.

Recall from Section §4.3 that we rely on passive and active measurements to uncover intermediaries, as well as on manual identification of crawlers and auctioneers, and lastly on a list of CDN servers. To account for the varying degrees of certainty when using these data sets, we distinguish between three different classes when quantifying back-office traffic. In particular, we consider back-office traffic caused by servers in our CDN data set (*IPs-CDN*), caused by servers we identified using DPI and manual inspection (*IPs-DPI+Manual*) and the back-office traffic caused by servers identified using the ZMap data set (*IPs-ZMap*).

Figure 4.3(a) shows the percentage of back-office traffic of the total Web traffic for each vantage point as a stacked bar plot. Thus, we depict the volume of back-office traffic found with the different methods: (a) information from the CDN only (the bottom bar), (b) information from the *IPs-CDN* and *IPs-DPI+Manual* (the sum of the bottom and the middle bar), (c) all information including ZMap (the sum of all bars). Across all vantage points we see at least 5% back-office Web traffic using the *IPs-CDN* and *IPs-DPI+Manual* set of IPs, confirming that back-office Web traffic is a significant contributor to today's Internet Web traffic. Even when only using the *IPs-CDN* data set, we see at least 4% back-office traffic at all vantage points except for L-IXP. This does not mean that the CDN is not active here but most of its traffic is front-office traffic. In terms of requests, the fraction of requests associated with back-office traffic is even larger with a minimum of 9% when using the *IPs-CDN* and *IPs-DPI+Manual* sets. This points out that some components of back-office traffic are associated with smaller transactions. But asymmetric routing—meaning the forward and return path do not use the same link— are likely the explanation for the extreme difference at BBone-1, where we see a huge number of back-office requests but only a relatively small percentage of back-office bytes. When we include the ZMap server IPs, the percentages of back-office traffic increases to more than 10% across all vantage points.

We next dissect the back-office traffic by type of activity using the *IPs-DPI+Manual* and the *IPs-ZMap* information. We illustrate our findings in Table 4.5, where we attribute back-office traffic to the entity that acts as client. We find that CDPs contribute 11% and 12% to the back-office requests and bytes in the L-IXP trace. The crawlers contribute 15% and 10% to the back-office requests and bytes, respectively. Surprisingly, the auctioneers are the big contributors to the number of requests with a share of 22% but



(a) % of Web traffic which is back-office across vantage points.

(b) % of Web traffic which is back-office over time (IPs-ZMap).

Figure 4.3: Fraction of back-/front-office Web traffic across vantage points.

Name	% of	CDPs	Auctioneers	Crawlers	Other
L-IXP	Bytes	12.1%	1.1%	10.3%	76.5%
	Requests	11.8%	22.5%	15.1%	50.6%
M-IXP	Bytes	73.3%	-	1.5%	25.2%
	Requests	65.7%	-	3.2%	31.1%
BBone-1	Bytes	50.7%	-	<0.1%	49.2%
	Requests	95.3%	-	<0.1%	4.6%
BBone-2	Bytes	93.6%	-	<0.1%	6.3%
	Requests	73.7%	-	4.3%	22%

Table 4.5: Classification of back-office Web traffic.

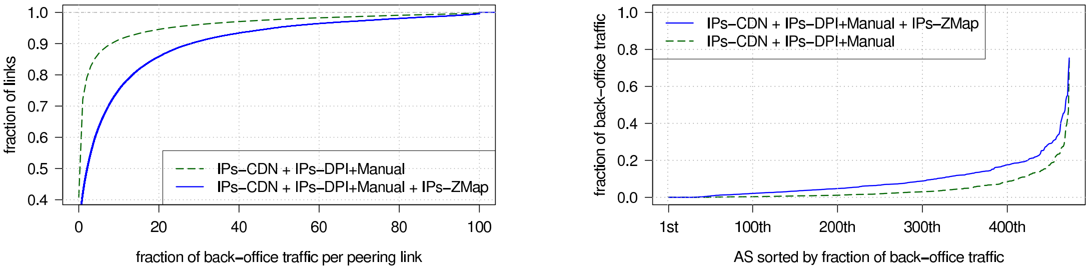
only 1% of the bytes. The rest of intermediaries contribute more than 76% and 50% of the back-office bytes and requests. The situation differs for the other vantage points, where CDPs clearly dominate the share of bytes and requests with at least 50% of the bytes and 65% of the requests.

Figure 4.3(b) shows how the percentages of back- and front-office bytes change over time using time bins of one hour. The percentages never decrease below 5% but can even exceed 40%. While some traffic is triggered by end-user action, activities such as crawling and cache synchronization are not. We see that, particularly for the two IXPs, the percentage of back-office traffic increases during the off-hours. The percentage of back-office traffic for BBone-2 increases on the third day of the trace by more than 10%. This increase may be due to (a) a routing change or (b) a change in the operation of the application infrastructure or (c) a change in the popularity of a Web application. In addition, we see more variability for the individual backbone links than for the IXPs. A likely explanation for this is that the IXPs aggregate the information from thousands of different peering links. Similar observations hold for the percentages of back-/front-office requests and responses (not shown).

4.4.3 Inter-domain traffic at an IXP

The two backbone traces illustrate that there can be notable differences in terms of percentages of back-office bytes and requests on different links, suggesting that links should be examined individually. Hence, we now take advantage of our ability to dissect the traffic seen on hundreds of individual AS-AS links at L-IXP.

Figure 4.4(a) shows the fractions of back-office traffic per AS-AS link (of the total traffic carried over it), where we sort them by the fraction of back-office Web traffic. We see that the fractions vary drastically



(a) ECDF: fraction of back-office traffic—per link.

(b) Fraction of back-office traffic—per member AS.

Figure 4.4: Back-office traffic across peering links at the L-IXP trace.

from 100% to 0%. Indeed, 18.2% (10.9%) of the peering links carry more than 15% (7%) back-office bytes when relying on the *IPs-ZMap + IPs-DPI+Manual* (*IPs-DPI+Manual*) data set. On the other hand, 25.5% (40.8%) of the peering links carry no back-office traffic at all. In order to get a better understanding of the most important AS-AS links, we inspect more closely the top-10 traffic-carrying links that have a fraction of back-office traffic larger than 95%. We find four links between cloud providers and content providers, three links between search engines and hosting providers, two links between CDNs and content providers, and one link between a content provider and an online advertisement company. This analysis illustrates the diversity of the players contributing to the back-office Web traffic.

If we aggregate the information to the level of IXP member-ASes, the overall observation does change a bit, as shown in Figure 4.4(b). We do not see member ASes that exchange only back-office Web traffic. They all have at least 20% front-office Web traffic. Nevertheless, most have some fraction of back-office traffic. There are 19.2% (18.0%) of the members with more than 15% (7%) back-office bytes for the *IPs-ZMap + IPs-DPI+Manual* (*IPs-DPI+Manual*) data set. Among the networks with the highest share of back-office traffic are cloud providers, hosting providers, a search engine, and an advertisement company.

4.4.4 Spatiotemporal characteristics

Temporal characteristics. To illustrate the temporal characteristics of some of the key players in the Web back-office, Figure 4.5 provides a time series plot of the number of requests seen at L-IXP and issued by content delivery proxies (CDPs), auctioneers, and crawlers, where we normalize the number of issued requests by the average number of crawler requests.

On the one hand, crawlers display rather constant activity throughout the week, which is the reason we use them for normalization. This constancy is to be expected because the request activity is not triggered by humans. The request patterns of the CDPs and auctioneers, on the other hand, display a diurnal pattern due to their connection to end-user activity. Interestingly, the rate of decrease between peak and off hours is larger for the auctioneers than for the CDPs. A possible explanation for the larger decrease is that the bidding process is a multiplicative factor of the end-user activity, i.e., one page visit triggers an auction involving multiple bidders. In terms of traffic volume (not shown), both CDPs and auctioneers exhibit typical diurnal variation while crawlers do not. While crawlers and CDPs dominate in terms of traffic contribution, auctioneers only contribute a tiny share of traffic. This is expected, as the bidding process involves numerous, but small, transactions.

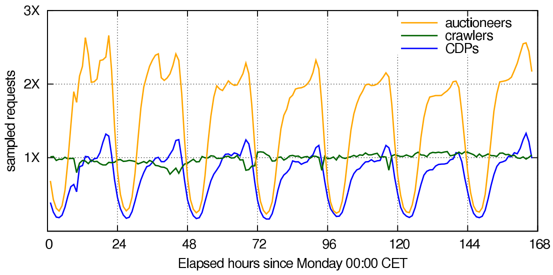


Figure 4.5: Time series: requests per hour by CDPs, auctioneers, crawlers (normalized by crawler requests).

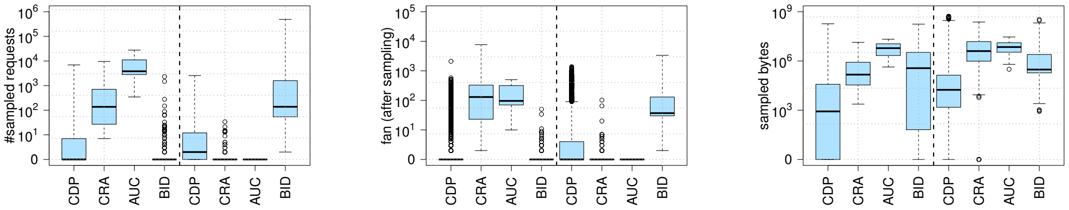
Spatial characteristics: Request forwarding. Noticing that many HTTP requests include the `Via` header, we next take a closer look at Web request forwarding. There are two HTTP header fields that are especially relevant for proxies: `Via` and `X-Forwarded-For`. The former indicates if a request has been forwarded through a proxy or a gateway; multiple `Via` field values represent each host that forwarded the HTTP request. The `X-Forwarded-For` field contains the IP address of the original client, followed by the IP addresses of the proxies on the path to the server. Thus, if present and correctly set, this header includes the original client IP as well as all proxy IPs that forwarded the request. This allows us to elucidate the complexity of the back-office by showing how far requests are forwarded and via how many Web proxies.

Inspecting these headers requires the ability to inspect the complete payload of a packet. We have full payloads for the BBone-1 and BBone-2 traces, and we extract from them the requests according to the previous *IPs-CDN+IPs-DPI+Manual* classification. Recall that a significant fraction of the requests in these traces are issued by IPs in *IPs-CDN*. Thus, the following analysis may be biased by the behavior of this particular CDN.

The `Via` header field indicates that while 12% of the requests traversed one proxy, another 77% traversed two proxies. We even observed a few requests that traversed seven proxies. With the `X-Forwarded-For` header field we now reconstruct the paths of the requests, i.e., for each request we extract the list of proxy IPs and append the destination IP at the end. Perhaps surprisingly, we find many private IP addresses among the IPs in the `X-Forwarded-For` headers, suggesting that either (a) end users use proxies on their premises and/or (b) proxies are located within data-center networks, where private IP address space is used. We argue that the second case dominates as the first IP in the list is typically publicly routable, e.g., belonging to an end user.

Out of the 1M requests we consider, we find 766K different client IPs appearing in the first position of the reconstructed paths. These IPs map to 7.9K different ASes, and around 94% of these IPs appear only once. The last IP in a reconstructed path may be an origin server or another proxy. We observe 5.9K different IPs appearing in this position, and they map to 350 ASes. Note that an observed request may be further forwarded. Finally, for the subset of IPs that do not appear at the beginning or end of a path (i.e., forwarding proxies), we find 16.5K different IPs scattered across 885 ASes. Notably, around 2.7K of the IPs are not publicly routable, yet they sum up to 40% of the occurrences.

To conclude this section, we take a look at the geographical characteristics of request forwarding. For this exercise we focus on the subset of requests detected via the *IP-CDNs* set. We then use validated information about the geographical coordinates of these CDN servers and rely on this CDN's commercial geolocation product to geolocate end users. We observe the following typical scenario for CDN activity, as seen from these backbone links: an end user issues a request to a front-end server (at 10 to 1000 km



(a) Sampled requests sent/received (left/right).

(b) Fan-out/in (left/right).

(c) Sampled bytes sent/received (left/right).

Figure 4.6: IP characteristics: Content delivery proxies (CDPs), crawlers (CRA), auctioneers (AUC), and bidders (BID) — L-IXP.

distance), this front-end contacts a back-office server within a CDN cluster (0 km distance). This back-office server in turn forwards the request to another back-office server that is still on the same continent (10-1000 km). Then, this proxy forwards the request to an origin server or to another back-office proxy across the Atlantic.

4.4.5 Communication patterns

Next, we return to examine the activity of the IPs in the L-IXP trace. Figure 4.6(a) shows, for the crawlers, auctioneers, bidders, and CDPs, a box plot of the number of sampled back-office HTTP/HTTPS requests we observed. Note that we only analyze back-office traffic characteristics here, and do not, for example, consider any requests related to the front-office activity of CDPs. We separate sent and received requests on the left and right sides of the plot. Accordingly, Figure 4.6(b) shows the observed fan-out (i.e., to how many different hosts were requests sent) and fan-in (i.e., from how many hosts were requests received). Figure 4.6(c) shows the number of sampled bytes received/sent.

Auctioneers and bidders. Auctioneers are the most active in terms of number of requests sent. From our sampled data we estimate that the average number of bid requests/hour issued by these IPs is roughly 232 million. This estimate implies that an average auctioneer IP issues more than 700K bid requests/hour. Overall, auctioneers also contribute significant numbers of bytes in both directions. Indeed, as Figure 4.6(c) shows, the number of bytes sent and received are of the same order of magnitude. This balance is reasonable given the underlying bidding protocol (e.g., [22]). Note, that the auctioneers only contact a limited set of servers, as highlighted in Figure 4.6(b). Correspondingly, the bidders are also contacted only by a limited set of auctioneers. However, in terms of received requests, not all bidder IPs are equally active – some of them receive just a few bidding requests while others see more than 450K sampled requests. Indeed, many bidders receive requests from different organizations simultaneously. Given the sampling ratio of this vantage point, we estimate that the most active bidders receive more than 42 million requests for bids per hour!

Crawlers. Crawler IPs are the second most active group of IPs in terms of requests sent. We estimate that, at this vantage point, crawling accounts for roughly 155 million requests/hour and that the most active crawlers issue up to 910K requests/hour. Naturally, the number of bytes received is larger than the number sent. Overall, we estimate that all together crawlers fetch roughly 3.8 TB per hour. However, not all are equally active, and we even see some fetching content from only a single IP.

Content delivery proxies. On average, the proxies show the lowest activity per individual IP. This observation applies to both bytes and traffic. However, due to their large number, they contribute significantly to back-office traffic. This category of IPs exhibits the largest variation in behavior, and some of the heavy hitters in this category compete with those in the other categories.

4.5 A CDN's perspective

Until now, we have analyzed back-office Web traffic from our vantage points in ISPs and IXPs. In this section, we present a complementary perspective provided by vantage points inside a commercial CDN. A CDN can be viewed as a high-bandwidth low-latency conduit that facilitates data exchanges between end users and different points of origin. As we have seen in §4.4.2, CDNs are one of the major contributors to back-office Web traffic on the Internet. This section delves into the details of a data set provided by a large commercial CDN to further characterize back-office Web traffic in the context of content delivery.

4.5.1 Dataset

Our analysis is based on server logs from the CDN's *edge*, or front-end, servers. Each log line records the details of an exchange of data where the edge server is one of the endpoints. Thus, the logs capture the interactions between the edge server and the end users, i.e., front-office Web traffic, as well as the interactions with other CDN servers and origin servers, i.e., back-office Web traffic.

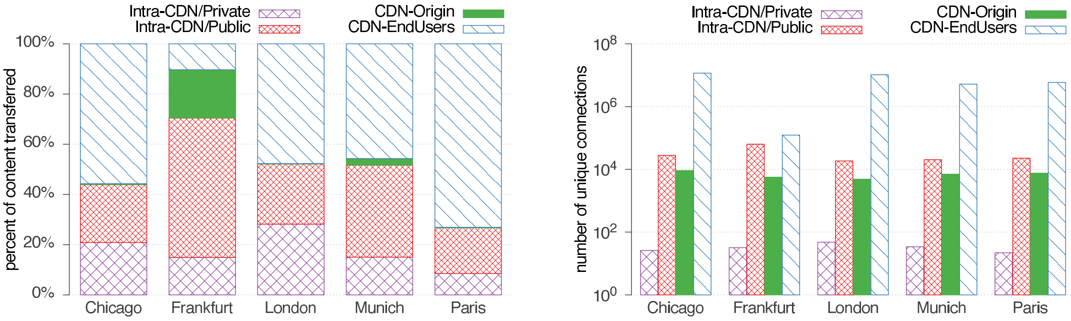
We obtained the server logs from all servers at one cluster in each of five different cities: Chicago, Frankfurt, London, Munich, and Paris.[5] Note that there may be multiple clusters at each city, and we selected only one of the larger clusters in each city. CDNs also deploy multiple servers at each cluster, e.g., for load-balancing, and servers at each cluster offer a diverse set of services ranging from Web-site delivery to e-commerce to video streaming. We selected clusters of servers configured to handle Web traffic, and our logs measure Web traffic of more than 350TB in volume.

4.5.2 Front-office vs. back-office traffic

The primary focus of a CDN is to serve content to the user as efficiently as possible. Therefore, one should expect CDN front-office traffic to dominate CDN back-office traffic in volume. As not all content is cacheable [66], up to date, or popular, some content has to be fetched from other servers. Moreover, many CDNs, e.g., Akamai [273], create and maintain sophisticated overlays to interconnect their edge servers and origin servers to improve end-to-end performance, to by-pass network bottlenecks, and to increase tolerance to network or path failures. Hence, a CDN edge server may contact, besides origin servers, other CDN servers located either in the same cluster, with back-office Web traffic routed over a private network, or in a different cluster at the same or different location, with the back-office Web traffic routed over a private or public network.

With the knowledge of the IP addresses used by the CDN's infrastructure, we can differentiate the intra-CDN Web traffic from the traffic between the CDN servers and end users (CDN-EndUsers), and CDN servers and origin servers (CDN-Origin). Furthermore, within the class of intra-CDN Web traffic, we can differentiate the traffic between servers in the same cluster from that between servers in different

[5] A small fraction of servers at each location did not respond to our requests to retrieve the logs, but this should not affect the analysis.



(a) Traffic volumes.

(b) CDN-EndUsers Fan in, Others Fan uut.

Figure 4.7: Front-office and back-office traffic at a large CDN.

clusters; traffic between servers in the same cluster uses high-capacity low-latency links and is routed over a private network (Intra-CDN/Private). We note that this traffic does not qualify as back-office Web traffic routed over the public Internet, which is the main focus of this chapter. But in order to properly account for the publicly-routed back-office traffic that we are interested in, we must be able to separate out the Intra-CDN/Private traffic. Note also that since this category of back-office Web traffic is not routed via the public Internet it does not accrue any peering cost or hosting cost. Our classification scheme partitions the Web traffic identified via the logs into four categories: (1) CDN-EndUsers, (2) Intra-CDN/Public, (3) Intra-CDN/Private, and (4) CDN-Origin.

Figure 4.7(a) shows the proportion of traffic observed in each of the above four categories at the five different locations (or clusters). Not surprisingly, we see that most traffic is, as expected, CDN-EndUsers traffic. We still observe at least 25% back-office traffic at each location. Of the five clusters, Paris is the most efficient from the perspective of the content provider, with more than 70% of the traffic in the CDN-EndUsers category, and CDN-Origin traffic very low (around 1%).

Frankfurt is an oddball. At Frankfurt, the end-user traffic accounts for less than 12%. After discussions with the CDN operator, we learned that servers in the Frankfurt cluster cache content from origin servers for other edge servers in nearby clusters. The high-volume of Intra-CDN/Public traffic (about 55%) is indicative of this role for the servers in the Frankfurt cluster. Besides reducing the latency involved in fetching the content from the origin servers, this practice limits the number of servers that have to fetch content from the origin servers. The traffic at other locations show significant volumes in both the Intra-CDN/Public and Intra-CDN/Private categories. These statistics are indicative of the reliance on cooperative caching with the CDN.

Recall from Section §4.4.4 that there is a wide range of diversity in the number of hops over which an HTTP request is forwarded, as well as the distances to the final server. Using the actual locations of each CDN server as ground truth, we computed the distances for all Intra-CDN data exchanges. Figure 4.8 plots the resulting ECDF of the distances for the Intra-CDN/Public traffic weighted by the content size. The cluster in Frankfurt, in addition to serving end-user traffic, acts as a caching hub, as explained previously. Figure 4.8 provides further evidence of Frankfurt's role as a caching hub. About 20% of the traffic to the cluster in Frankfurt is being transferred over trans-continent links.[6] Contrast this with the cluster in Munich which receives around 2% of it's intra-CDN traffic via trans-continent links; discussion with the CDN operator confirmed that Munich does not serve as a caching hub. Figure 4.8 also reveals that a substantial fraction of the traffic travels only a short distance. This is expected, since

[6] We assume that distances of 6000 km or more indicate trans-continent links.

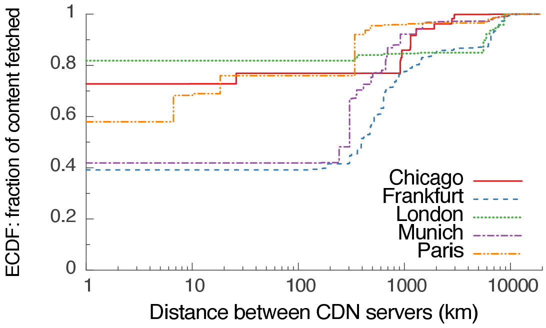


Figure 4.8: Content volume by distance (Intra-CDN) at a large CDN.

in large metropolitan areas, like those selected for our study, edge servers are located at multiple data centers in the same city.

4.5.3 A deeper look at the CDN's back-office

Previously, we observed that the hosts' fan out, i.e., the number of hosts contacted by a host, can vary significantly. Accordingly, we may ask if fan out varies among the different classes of back-office CDN traffic. Not surprisingly, it turns outs that the number of unique end-user addresses to which the edge servers deliver content, i.e., the fan in, is larger than the combined number of CDN and origin servers from which they fetch content, i.e., the fan out.

Figure 4.7(b) shows the number of unique connections observed in the different traffic categories at each location. From the figure, we see that the number of unique connections in the back-office traffic categories (Intra-CDN/Private, Intra-CDN/Public, and CDN-origin) is two orders of magnitude less than that in the CDN-EndUsers category; note that the y-axis is plotted using a log scale. Moreover, the Intra-CDN/Private category mainly captures intra-cluster data exchanges and thus the fan out is even smaller. Finally, although the number of unique connections in the CDN-Origin category is smaller, it is equivalent in order of magnitude to the connection count in the Intra-CDN/Public category.

Aggregating the traffic volume by server addresses in both the CDN-Origin as well as the Intra-CDN/Pu blic category reveals that the traffic is not uniformly distributed across all servers; rather there are heavy hitters. 20% of the origin servers contribute to more than 96% of the content delivered to the CDN's edge servers. A similar trend manifests in the Intra-CDN/Public category; 20% of the CDN's servers account for over 94% of the traffic volume moved from different servers in the CDN's infrastructure to the front-end, or edge, servers. These figures hint at the impact of the varying popularity and cacheability of content on the traffic patterns within the CDN infrastructure.

4.6 Summary

The Web and its ecosystem are constantly evolving. In this chapter we realized a first step towards uncovering and understanding one component of this ecosystem that is increasing in complexity, but remains understudied: back-office Web traffic. There is an entire ecosystem of automated infrastructure necessary to support how users interact with the Web today. This infrastructure, and by extension the traffic it generates, supports Web content delivery, monetization and search. By using a diverse set of vantage points, we have shown that back-office traffic is responsible for a significant fraction not only of today's Internet traffic but also today's Internet transactions.

We find a significant percentage of back-office Web traffic in our traces and that it varies from vantage point to vantage point. Indeed, the back-office traffic carried over the backbone links is mostly dominated by content delivery proxies (CDPs). The picture differs when looking at IXPs, where we can monitor multiple links at once. While most of the back-office traffic there is also due to CDPs and other intermediaries, real-time-bidding and crawling also contribute a significant share of bytes and Web requests. Our analysis illustrates that a significant part of the back-office traffic is not triggered by end users. We see that besides the expected players such as CDNs, search-engines and advertisement companies, cloud service providers and Web hosting providers are also responsible for a large fraction of back-office Web traffic. In fact, these companies provide the resources needed to build complex and scalable Web services, which often require communication between multiple servers to operate. Our analysis illustrates how back-office traffic is present in multiple links on the AS-level, revealing a complex ecosystem of players in support of the Web.

In more detail, we observe that real-time bidding is very prominent and relies on many small transactions involving a fairly small set of organizations and hosts. As each end-user request may trigger multiple bid requests, RTB significantly contributes to the number of back-office transactions. Crawling, on the other hand, happens on a coarser-grain time scale and is executed by a limited number of organizations that constantly fetch content from a diverse set of mainly Web hosting providers. Finally, while CDPs have a diverse profile, our analysis illustrates that a single end-user request to a CDN front-end server can involve a chain of proxies. These connections remain entirely hidden to the end users. We confirm our observations using data from a large CDN and further characterize back-office Web traffic at this service provider.

At the same time we showed that back-office traffic is responsible for a significant fraction of today's Web traffic, we offer two key reasons to explain this phenomenon: *i)* sustained deployment of front-end servers close to end users, and *ii)* substantial data exchanges conducted by service providers as part of their operations viz., coordination and synchronization of components of their distributed infrastructure. With the increasing trend to offload "computations" to the cloud to support enterprise applications [270] or desktop application, e.g., the new release of Microsoft Office, Apple iCloud, and DropBox [133], it is clear that the volume of back-office Web traffic will continue to increase. First, application servers at different data centers will have to exchange data with each other for data synchronization and replication. Second, as edge servers attempt to offer more services [145, 206], viz., personalized Web experiences for end users, they will need to communicate with other edge servers and back-office servers, at different locations, to retrieve relevant content for the end users.

In Chapters §3 and §4 of this dissertation we have studied how three enablers for today's Web –*content delivery, monetization, and search*– manifest in Internet's traffic, i.e., as front- and back-office Web traffic. From this perspective, Web usage is subject to network conditions, e.g., connectivity loss, congestion. Operational decisions concerning the Internet's infrastructure and protocols can thereby impair how users interact with the Web. In the next part of this dissertation we study two aspects thereof: *i)* QoE degradations due to over-sized packet buffers, and *ii)* the Internet's transition to IPv6.

Part II

Using the World Wide Web: Impact by Internet's infrastructure

5 Connectivity: Transition to IPv6

The past two chapters of this thesis are centered on three fundamental enablers for today's Web: *i)* content delivery, *ii)* search: crawling to index content, and *iii)* monetization via advertisements. In particular, we show how these functions result in traffic at multiple vantage points on the Internet. In turn, given that Web is a distributed system that runs on top of the Internet [286], decisions regarding Internet infrastructure can alter the way users interact with this information system. We investigate one of them in this chapter: the Internet's transition to IPv6.

The Internet Protocol (IP) is the Internet's network layer protocol. The initial and widely deployed version 4 of the Internet Protocol has a fundamental resource scarcity problem (see Section §2.2.1). IPv6, which offers a vastly larger address space was intended to replace IPv4 long before the scarcity of IPv4 address blocks began. However, after almost two decades of IPv6 development and consequent efforts to promote its adoption, the current global share of IPv6 traffic still remains low. Urged by the need to understand the reasons that slow down this transition, the research community has devoted much effort to characterize IPv6 adoption, i.e., *if* ISPs and content providers enable IPv6 connectivity. However, little is known about *how much* this is actually used, i.e., which factors determine if two parties exchange data over IPv4 instead of IPv6.

Determined to investigate the reasons that refrain the increase of IPv6 traffic on the Internet, we tackle a smaller but more feasible challenge and study this problem from the perspective of 12.9K subscribers of a dual-stack ISP. This vantage point presents a unique opportunity for analyzing the interactions between subscribers and service providers, and how they influence the share of IPv6 traffic. Our main findings are the following:

1. Even though this ISP supports IPv6 connectivity, a large number of subscribers are not yet prepared for IPv6. While in some cases it is the ISP which does not provide IPv6 connectivity to its subscribers, more often the CPE limits IPv6 connectivity.
2. Consequently, IPv6-ready services exchange a significant amount of traffic over IPv4. IPv4-only speaking devices and fall-back mechanisms further increase the share of IPv4 traffic for these services. Nonetheless, we observe a strong *intent* for IPv6 traffic that IPv4-only services are not yet ready to correspond to.

3. Due to dual-stack applications' preference for IPv6, dual-stack networks have a high risk to experience a rapid and substantial increase of the IPv6 traffic share if a few major service providers enable IPv6.

The rest of this chapter is organized as follows. We describe our methodology and data set in §5.2 and in Section §5.3 we present our findings. We discuss the implications and limitations of our work in §5.4.

5.1 Background

The initial and widely deployed version 4 of the Internet Protocol has a fundamental resource scarcity problem: it reached the limit of available, globally unique, IP address space. As of today, IPv4 address scarcity has become a global issue, forcing some ISPs to NAT large chunks of their customers or even to buy blocks of remaining free IPv4 address space on address markets [212,253]. IPv6, which offers a vastly larger address space was intended to replace IPv4 long before the scarcity of IPv4 address blocks began. However, despite initiatives by Internet governing bodies to promote IPv6 deployment [59], the transition and deployment of IPv6 has been slow and challenging in production environments [61,118]. As of today, there is no clear consensus about when IPv6 will really "hit the breaking point", i.e., when IPv6 will become the preferred interconnectivity option on the Internet.

The research community has put substantial effort into measuring and tracking IPv6 deployment with the goal of assessing this transition (e.g., [122]). Yet, anecdotal claims like *IPv6 is here already* to *IPv4 will stay forever* are typically motivated by two different measures of IPv6 adoption: *connectivity* and *traffic share*. For example, while Google reports optimistic *connectivity* adoption rates as high as 11% for end hosts [56], the IPv6 *traffic share* at major Internet eXchange Points (IXPs) still ranges between 1-2% [54]. This global low share of IPv6 traffic is not just the main cause for disappointment regarding the pace of IPv6 adoption, but it has also fueled a different interconnection structure among ISPs. The provider hierarchy in the IPv6 Internet shows vastly different properties compared to that of IPv4 [158], i.e., the one ISP offering free IPv6 tunnels has the largest customer cone in the IPv6 Internet, whereas Tier-1 ISPs with worldwide backbones are less prominent in this hierarchy. We refer the reader to Section §2.2.1 for a summary of related work on *IPv6-adoption*.

5.2 Methodology and dataset

There is, indeed, a fundamental mismatch between IPv6 connectivity statistics and IPv6 traffic share. To exchange data over IPv6, all components on the path from a source to a destination need to fully support IPv6 (see Figure 5.1). This includes *(i)* end-user devices and operating systems supporting IPv6, *(ii)* applications making proper use of the available connectivity options (see [287]), *(iii)* customer premises hardware (CPEs) supporting and providing IPv6 to the home network [55,173,271], *(iv)* the ISP assigning IPv6 to the subscribers CPEs [128], and finally *(v)* content providers enabling their services over IPv6 [219].[1] Moreover, even if all of the above conditions apply, i.e., all components *support* IPv6, a second dimension of the problem is whether IPv6 will be preferred over IPv4 as modern applications employ a technique named "*happy eyeballs*" to *choose* between IPv4 and IPv6 according to the current network conditions [304].

Hence, a dual-stack ISP presents a unique opportunity to study the interactions of this ecosystem and its influence on the share of IPv6 traffic. Central to our work is to identify why some octets of data are

[1] We consider only *native IPv6 traffic* in this work, i.e., we do not consider tunneling approaches such as *Teredo* (see, e.g., [262]).

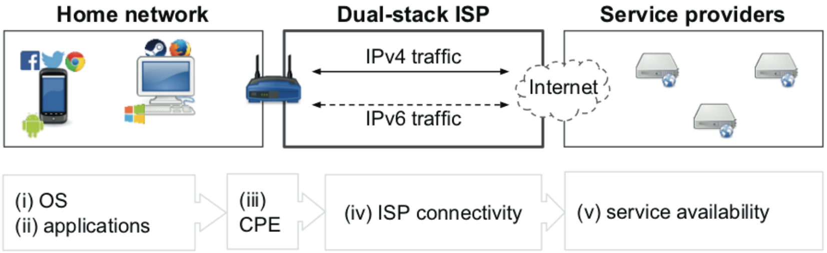


Figure 5.1: IPv6 traffic in dual-stack networks. Barriers are present at home networks (operating systems, applications and CPEs), ISPs (offered DSL connectivity), and at service providers.

encapsulated with IPv4 or IPv6 headers. To this end, we first need to discover the connectivity options of the two engaged parties. Only with this information at hand we can reason about traffic exchanged by dual-stack subscribers and services.

5.2.1 Measuring IPv6 connectivity

Customer connectivity. Broadband network providers typically rely on Remote Authentication Dial-In User Service (Radius) to assign IP addresses to subscribers [255]. With this protocol, CPEs obtain IP addresses, usually a single IPv4 address that multiplexes devices (NAT). This protocol specification also supports the delegation of IPv6 addresses to subscribers [62, 128, 257]. If the CPE gets assigned an IPv6 prefix, we say that the subscriber obtains IPv6 connectivity from the ISP. The raw traffic statistics tell us later whether the subscriber's devices make actual use of this IPv6 prefix.

Since not all devices within home networks support IPv6, the raw traffic statistics are necessary but not sufficient to infer if a device within a subscriber's premise can use IPv6. We use AAAA DNS requests as an indicator for the presence of IPv6-speaking devices. Most dual-stack applications follow the *happy-eyeballs* proposed standard (see [304]), and issue A as well as AAAA DNS requests. If the requested service is available over IPv6 the device attempts to connect simultaneously to two addresses contained in the DNS resource records (RRs); one being IPv6 and the other IPv4. An application that adheres to the example implementation establishes then two TCP connections and uses the one that completed the handshake faster. Some implementations introduce a preference towards IPv6. For example, Apple devices issue an IPv6 connection immediately after a successful AAAA request if the A response did not arrive already, or historical RTT data suggests a difference > 25 ms [264]. Given that most DNS clients issue AAAA requests first [227], some dual-stack devices do not always attempt a connection on IPv4 and IPv6 although they issue requests for both RRs.

One important fact regarding IPv6-speaking devices is that many resolver libraries avoid suppressing AAAA requests if there is no global —but just link-local—IPv6 connectivity. The rationale is that doing so can lead to undesired situations [11]. Thus, we can use this information to further identify CPEs that offer link-local IPv6 connectivity even if the ISP does not provide IPv6 connectivity to them.

Service connectivity. Assuming that the network that hosts a service supports IPv6 then a content provider can make its service available over IPv6 by just updating its DNS AAAA and PTR resource records (RRs) [219]. Henceforth, we can analyze DNS traffic to infer if a service is IPv6-ready by looking for non-empty AAAA responses. However, as we may not be able to observe all AAAA RRs (e.g., if the clients are not IPv6 enabled), we complement passive data with active measurements.[2]

[2] We conducted these additional measurements shortly after the data collection.

5.2.2 Measuring IPv6 usage

Once we are aware of the connectivity options of a service provider, we associate its services to traffic. The main idea is to associate the DNS requests issued by an IP address to the network flows it generates, i.e., reproduce the mapping between host names and server IPs for each subscriber. This problem has been already explored (see, e.g., [90, 226, 240]). It is important to notice that for dual-stack networks the IP addresses of the flows and those of the DNS traffic are not necessarily the same. Therefore, we cannot directly use the source IP of a DNS request as a *rendezvous*. Instead, we keep track of the IPv4 and IPv6 addresses assigned to each subscriber. Another caveat (as reported in related work) is that we need to update this mapping according to the TTL values of the DNS response RRs. We are aware that related studies have reported violations of the TTL field by clients [106, 226]. For example, Callahan et al. [106] observe that 13% of the TCP connections use expired addresses and attribute it to security features present in modern Web browsers. In this work we opt for a conservative approach and strictly use the TTL expiration values. In addition, we do not consider negatively cached responses, e.g., a service without a AAAA RR. Our rationale is that although negative answers should, in principle, be cached according to the SOA record [71], some resolvers do not respect this [204]. The immediate consequence is that at times we will not observe a AAAA request for services without AAAA RR and may mis-attribute it to devices that do not support IPv6.

Annotation of flows. Following the aforementioned approach gives us the ability to annotate network flows with the following information: i) the ISP has delegated an IPv6 prefix to the subscriber's CPE, and when possible also ii) the Fully Qualified Domain Name (FQDN) associated with the flow, and iii) the subscriber issued an A and/or a AAAA DNS request. After collecting the trace we extend this annotation with the following information: iv) the subscriber makes use of the IPv6 prefix, and iv) the connectivity options for the FQDN, i.e., the service is available over IPv4 and/or IPv6.

5.2.3 Dataset

Having described our methodology, we next describe how the trace was collected and annotated. We employ a custom tool built on top of the *libtrace* library [68] to produce two streams of data from raw network data. The first stream consists of packet summaries. In particular, for every IP packet, this tool reports the packet size, the protocol number, the SRC and DST IP addresses, and where applicable, the port numbers. For TCP packets, it also records the flags (e.g., SYN), the SEQ, and ACK numbers. The second stream consists of full-sized packets of DNS traffic (UDP port 53). At the same time we process this stream, we obtain from the first one flow-level statistics. Namely, we aggregate the packet summaries into the 5 tuple and expire inactive flows after 3600 s. For TCP flows we also compute the time difference between a SYN packet and the packet acknowledging it.[3] Given the location of our monitor within the aggregation network, these "handshakes" only the capture wide-area delays (backbone RTTs) and they do not include delays introduced by the access and the home network. Finally, we remark that the data set was collected, processed, and analyzed at an isolated and secured segment infrastructure of the ISP. The toolset operates in an automated fashion and anonymizes line ids and addresses before writing the annotated flows to the disk. Table 5.1 summarizes the dataset collected for this study.

DNS transactions. We process 141.9M DNS transactions, each transaction being an A or a AAAA request and a valid response.69.6% of these entries are of type A. 0.6% and 36.0% of the A and -respectively- AAAA could not be resolved (empty response). The highest ratio of AAAA is because some content is still not accessible over IPv6 (see §2.2.1). 39% of the A requests were over IPv6, and 28%

[3] We exclude from this computation flows accounting with retransmissions of packets with SYN flag set.

Trace	Total	TCP$_{v4}$	TCP$_{v6}$	UDP$_{v4}$	UDP$_{v6}$
#bytes	64.5T	80.5%	10.7%	7.4%	1.1%
#flows	356.2M	53.1%	4.7%	18.2%	21.7%

Description: 45 h trace collected during winter 15/16 (12.9K DSLs)

Table 5.1: Dataset overview. Dissection of IP traffic by IP version and transport protocol

of the AAAA over IPv4. 91.1% of the requests are routed to a name server within the ISP (53% of them over IPv6). The rest is distributed among different DNS providers.

Flow-level statistics. Table 5.1 shows the overall contribution of UDP and TCP traffic by IP version. Unsurprisingly, TCP$_{v4}$ dominates in terms of traffic volume. However, the share of IP$_{v6}$ is substantial (11.9%) specially when compared to older measurement studies at other vantage points [122,262]. Web traffic sums up to 86.6% of the trace volume (13.5% over IPv6).[4] We find that QUIC contributes 2.8% of the overall trace volume (39.5% over IPv6). In terms of flows percentage, the share of UDP$_{v6}$ flows is well above the UDP$_{v4}$ share. This is a bias introduced by DNS traffic: in this vantage point DNS accounts for 71.0% of the UDP flows (75.3% of it over IPv6).

Classification coverage. We are able to associate up to 76.1% of the traffic to services (84.1% if we consider just Web traffic). While our coverage statistics are consistent with the base results reported in [226], we remark that ours are lower than related methods because our method *i)* does not use a warm-up period to account for already cached DNS RRs, *ii)* relies on each subscriber's own DNS traffic, and *iii)* adheres to the TTL values included in DNS responses.

5.3 A dual-stack ISP perspective

5.3.1 Subscribers

We can find three classes of DSLs among the 12.9K subscriber lines of this vantage point: *i) IPv4-only*: lines that do not get IPv6 connectivity from the ISP (17.3%), *ii) IPv6-inactive*: lines provisioned with IPv6 connectivity but no IPv6 traffic (29.9%), and *iii) IPv6-active:* lines with IPv6 connectivity as well as IPv6 traffic (52.9%).

***IPv4-only* subscribers.** This set of lines corresponds to subscribers for which the ISP has still not activated IPv6 connectivity (e.g., old contracts). They contribute to 12.0% of the overall trace volume. 26.6% of their traffic is exchanged with services that are available over IPv6. We notice that some devices issue AAAA DNS requests, most likely because some CPEs create a link-local IPv6 network. In fact, for 11.6% of the traffic related to IPv6 services we observe a AAAA request. This first observation is relevant for *IPv6-adoption* studies, as it indicates that in some cases DNS traffic may not well reflect the actual connectivity. In fact, it underlines that many devices are already prepared to use IPv6 connectivity waiting for the ISP to take proper action.

***IPv6-inactive* subscribers.** For 36.1% of the DSLs we do not observe any IPv6 traffic, despite the CPEs get assigned IPv6 prefixes. One explanation is that some CPEs provision IPv6 connectivity, but *i)* IPv6 is not enabled by default for the local network, or *ii)* the user disabled it. Other (less likely) explanations are that none of the devices present at premises during the trace collection support IPv6 (e.g., Windows XP), or they do not contact services available over IPv6. This is unlikely, as 24.1% of the traffic is exchanged with IPv6-ready services. We find that only 1.7% of the traffic can be associated

[4] TCP traffic on ports 80 and 8080 (HTTP), 443 (HTTPS), as well as UDP traffic on port 443 (QUIC).

Service	**Subscribers**			**Sum**
	IPv4-only	*IPv6-inactive*	*IPv6-active*	
IPv4-only	5.4%	20.1%	22.4%	47.9%
IPv6-ready	3.2%	9.2%	15.4%	27.8%
IPv6-only	0.0%	0.0%	< 0.1%	< 0.1%
Unknown	3.4%	8.8%	12.1%	24.2%
Sum	11.9%	38.1%	49.8%	

Table 5.2: Traffic contribution per subscriber and service provider according to their connectivity. In contrast to *IPv6-active* DSLs, *IPv6-inactive* also provision IPv6 connectivity but do not generate IPv6 traffic.

with a AAAA request, probably because most devices suppress AAAA requests in the absence of a link-local IPv6 address. Hence, in contrast to the previous category of subscribers, we cannot estimate the amount of traffic that could be exchanged over IPv6 once the ISP grants them IPv6 connectivity.

***IPv6-active* subscribers.** The share of IPv6 traffic for the rest of subscribers, those with IPv6 connectivity and traffic, is almost twice (21.5%) that of the overall trace. If we only consider traffic exchanged with services available over IPv6 the ratio is even higher (69.6%) probably because happy eyeballs dominate and prefer IPv6. Indeed, 85.1% of the traffic exchanged with these services and annotated with an A and a AAAA request is IPv6. This is important for service providers and operators, as it implies that enabling IPv6 can increase the share of IPv6 traffic from/in dual-stack networks rapidly.

5.3.2 Service providers

We next shift our focus to service providers. For 76.1% of the traffic that we can associate to services, 63.2% is available only over IPv4, 36.6% over both operational versions of the IP protocol, and the remainder *apparently* only over IPv6. Accordingly, we report in Table 5.2 the cross-product of the different subscriber and service provider categories and their corresponding contribution to the traffic. In the following, we elaborate on the characteristics of these three classes.

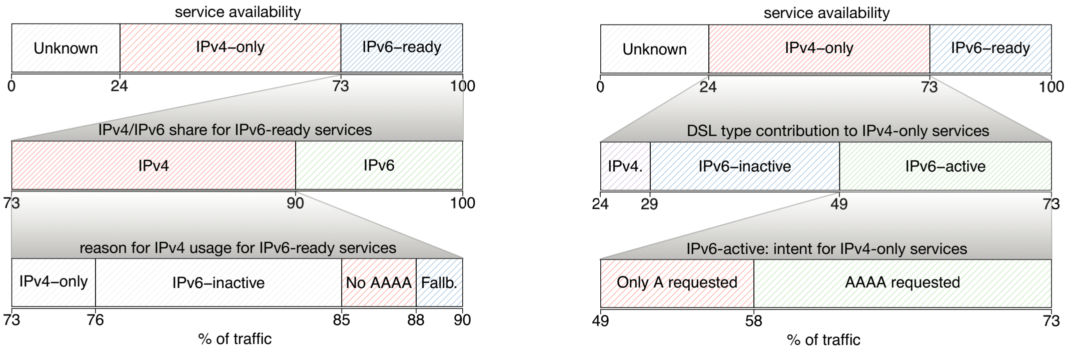
***IPv4-only* services.** As expected, this set of services dominates the share of traffic (47.9%). However, for 36.2% of this traffic, we observe a preceding AAAA request, which implies that traffic has the potential to be served over IPv6 if the corresponding providers enable IPv6.

***IPv6-only* services.** We find around 500 services that *appear to be* available only over IPv6. They account for less than 0.1% of the traffic. Manual inspection of these services reveals that most of them are connectivity checkers. In addition, some service providers add specific strings to the host names, which may appear to us as an IPv6-only service, e.g., both *host.domain.org* and *hostv6.domain.org* have a AAAA RR, but only the former has an A RR.

***IPv6-ready* services.** These services generate a non-negligible amount of traffic (27.8%). However, having learned that many subscribers from this dual-stack network cannot use IPv6 it is not surprising that the actual share of IPv6 traffic for these services is “just” 38.6%.

5.3.3 IP traffic

Table 5.2 reveals two upper bounds for the IPv6 traffic share. While *IPv6-active* subscribers contribute to 49.8% of the overall trace traffic, *IPv6-ready* services account for 27.8%. Not only these two percentages



(a) **IPv6 barriers.** Top bar: service availability. Middle bar: used IP version. Bottom bar: traffic from subscribers without IPv6 connectivity (*IPv4-only* and *IPv6-inactive*), and traffic from *IPv6-active* subscribers (traffic without a AAAA and happy eyeballs falling back to IPv4).

(b) **IPv6 intent.** Top bar: service availability. Middle bar: subscriber category. Bottom bar: traffic from *IPv6-active* subscribers to *IPv4-only* service providers (% of traffic associated with a AAAA DNS request).

Figure 5.2: Barriers and intent for IPv6 traffic in a dual-stack ISP.

are well above the actual trace's IPv6 share (11.9%), but also, their cross product is above it (15.4%). Fueled by these observations, we proceed to study the root causes that lead to this situation.

IPv6 barriers. Figure 5.2(a) illustrates why traffic related to *IPv6-ready* services is exchanged over IPv4. On the top of the Figure, we show a bar regarding all traffic in the trace according to the service availability. As previously stated, 27.8% of the traffic relates to services available over IPv6. Nevertheless, most of it (61.4%) is actually exchanged over IPv4 (see middle bar). With the third bar we illustrate why data is exchanged over IPv4 instead of IPv6. Most of this traffic (70.5%) is IPv4 because the subscribers do not use IPv6 connectivity (*IPv4-only* and *IPv6-inactive*). We make two observations for the remainder of this traffic (which is generated by *IPv6-inactive* subscribers). Most of it has no associated `AAAA` request. This can be for two reasons. First, the device does not support IPv6. Second, the application did not respect the TTL (see §5.2.2). For 40% of the IPv4 traffic from *IPv6-active* subscribers to *IPv6-ready* services we observe a `AAAA` request. These are likely *happy-eyeballs* falling back to IPv4.

IPv6 intent. Figure 5.2(b) illustrates how even though many devices are IPv6 ready, the service providers cannot supply the demand of IPv6 traffic. We focus on traffic from *IPv4-only* services (top bar). While the bar in the middle depicts how much of this traffic they exchange with each subscriber category, the bottom bar shows the traffic characteristics for the *IPv6-active* subscribers. In particular, we observe that clients issue `AAAA` requests for 62.5% of this traffic. We remark that the remainder of this traffic is not necessarily generated by *IPv4-only* speaking devices, as negative caching for long periods of times may bias our observations (§5.2.2).

Happy eyeballs. Having realized that parts of IPv4 traffic can be attributed to (un-)happy eyeballs, we now study two metrics concerning dual-stack applications and devices, i.e., the RTT estimates and the DNS resolution times (see [264]). Our RTT estimate corresponds to the backbone RTTs (§5.2.3). For the DNS resolution time, we only consider transactions with non-empty responses and for which there are just one request and one response in the same UDP flow. We aggregate these per host name and compute the median only for those host names with at least 10 samples. We plot the ECDF for both metrics in Figure 5.3. Generally, dual-stack services offer similar conditions, i.e., around 80% of the values are within a range of 10 ms. Under such conditions, happy-eyeball implementations likely select IPv6, as

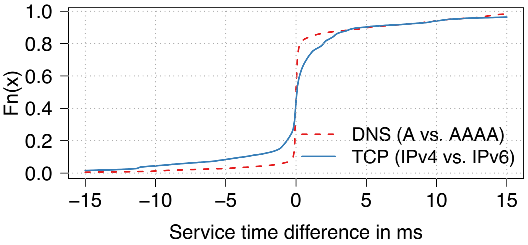


Figure 5.3: ECDF for the difference between IPv6 and IPv4 TCP hand shake and DNS resolution times per host name. Positive values indicate that IPv6 transactions took longer than those over IPv4.

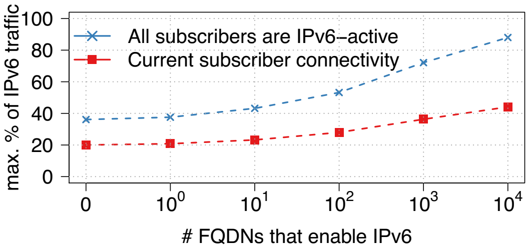


Figure 5.4: Estimation of the maximum *possible* share of IPv6 traffic when IPv4-only FQDNs enable IPv6. We sort FQDNs by their contribution in terms of bytes and do not take into consideration unclassified traffic.

indicated by our earlier results. This observation is important for service providers transitioning their services, as it implies that with proper provisioning for IPv6 they can expect a significant increase of IPv6 traffic if dual-stack consumer networks create most of their traffic. We also note that the final *choice* of connectivity is subject to how different implementations adapt to network conditions [60,178,179].

5.3.4 Case studies

We next describe two case studies: a large content provider and a large CDN, which nicely illustrate two opposite facets of the transition to IPv6. These providers contribute together to 35.7% of the overall and 73.1% of the IPv6 traffic. Both providers operate various Autonomous Systems (ASNs) as well as caches inside ISPs. To identify their traffic we rely on the origin ASN as derived from the IP addresses in the flows. To identify traffic from caches, we obtain a list of the Fully Qualified Domain Names (FQDNs) associated with IPs managed by these ASNs. We then search for flows within the ISP ASN that have associated FQDNs in the list.

A large content provider. Our first case study is a content provider that actively supports and promotes IPv6. 37.6% of its traffic is IPv6, and it alone contributes 69.9% of all IPv6 traffic in the trace. After annotating 91.8% of the traffic with FQDNs, we corroborate that almost all content requested by users at this vantage point is available over IPv6 (98.7%). *IPv4-only* and *IPv6-inactive* subscribers generate 74.1% of the IPv4 traffic while the share of IPv6 traffic for the *IPv6-active* subscribers is 70.5%. This observation suggests that for this content provider the connectivity of the subscribers is the main obstacle for the increase of IPv6 traffic.

A large CDN. We annotate 84.7% of this CDN traffic with FQDNs, and find consistent behavior for data exchanged between *IPv6-ready* services and *IPv6-active* subscribers: the share is in this case 83.4%. However, only 2.5% of the traffic is IPv6; as only 3.3% of this CDN traffic relates to *IPv6-ready* services. This implies that here the bottleneck for IPv6 is the server-side, since only 2.1% of the content requested with a AAAA is actually exchanged over IPv6.

Transition to IPv6. Service providers willing to transition their services to IPv6 need to update the corresponding DNS RRs. To illustrate the potential impact of this process on the share of IPv6 traffic, we next concentrate on *IPv4-only* services. We present in Figure 5.4 an upper bound for the share of IPv6 traffic when the top contributing FQDNs in terms of traffic enable IPv6. We produce two estimates. The first one assumes that there are no changes in the subscribers connectivity. The second one assumes that all subscribers become *IPv6-active*. Note, we do not take into consideration 24.2% of the bytes in the trace as we cannot associate them with a service. Enabling IPv6 connectivity for all subscribers immediately doubles the upper bound for the IPv6 traffic share; but most important is that if 10K services enable IPv6 at the same time, almost all data could be exchanged over IPv6. However, as described earlier in this chapter, *IPv4-only* devices and *happy-eyeballs* fall-backs to IPv4 can reduce this share.

5.4 Summary

We are very well aware that our vantage point is not representative of the Internet as a whole. While this particular ISP promotes IPv6 connectivity, others opt to additionally deploy Carrier Grade NATs to face IPv4 addresses scarcity. However, we argue that our observations most likely apply to other dual-stack ISPs (e.g., [167]) as well. Hence, these observations can help operators and providers by providing guidance on how to provision for IPv6 or insights on traffic dynamics during this phase. For example, *IPv4-only* service providers could exchange up to 30% of their traffic over IPv6 if they enable IPv6. By contrast, although 53% of the IPv4 traffic to *IPv6-ready* services involves subscribers whose CPE most likely does not provide IPv6 connectivity to its home network, *happy eyeballs* usually *choose* IPv6 over IPv4 (85%). We posit that IPv6 traffic shares will be subject to sudden increments in the advent of virtualization techniques at the edge of the Internet [49]. Virtual CPEs [108] will make easier for operators to transition their subscribers to IPv6 and troubleshoot IPv6-related problems. Avenues for future work include a closer investigation of issues specific to devices and applications as well as a characterization of *happy-eyeballs* fall-backs to IPv4.

The fundamental importance of the Internet's transition to IPv6 is strongly reflected in the continuous efforts within the research community to measure *IPv6 adoption* across the Internet. In this work, we make a step further and study a less-known aspect thereof: *IPv6 usage*. We reveal obstacles hampering IPv6 traffic in dual-stack ISPs, including applications falling back to IPv4, CPE configurations or the broad lack of IPv6 support among service providers. In spite of such obstacles, we report a pronounced increase, intent, and potential for growth regarding IPv6. We expect that the increasing IPv6 traffic shares will eventually lead to provision proper IPv6 infrastructure, establish genuine interconnectivity, and finally make IPv6 the first-class citizen on the Internet.

Our results show that a significant fraction of Web services are not available over IPv6, which has two consequences. First, native IPv6 clients need to rely on mechanisms that grant them the ability to reach Web content that is available only over IPv4. Second, as content providers transition to IPv6, subscribers in dual-stack networks may experience different network conditions when browsing the Web, depending on which IP protocol version they use. This observation motivates the next chapter: how large delays introduced by over-sized buffers impair the Quality of Experience of an end-user.

6 Transport: Impact of buffering on QoE

Improving Web performance has triggered the deployment of highly complex distributed systems on the Internet. See for example the CDNs as described in Section §2.1.1. Sometimes these optimizations can not help to counter certain bottlenecks, e.g., at the congested links of an under-provisioned access network. While ISPs rely on traffic engineering to reduce the load on their networks and thereby improve end-users Quality of Experience (QoE), network infrastructure components still require mechanisms to mitigate transient congestion originating from TCP's congestion control design. Packet buffers are, to this end, widely deployed in routers. Their purpose is to "absorb" transient bursts and thereby reduce packet loss at the cost of higher latencies.

Despite decades of operational experience and focused research efforts, standards for sizing and configuring buffers in network systems remain controversial. An extreme example of this is the recent claim that excessive buffering *–buffer bloat–* can severely impact Internet services e.g., the Web browsing experience. The goal of this chapter is to evaluate the impact of buffer-sizing choices on QoE to pave the way for more informed sizing decisions.

Unlike previous studies that consider Quality of Service (QoS) metrics (e.g., packet loss or throughput) our study focuses on end-user Quality of Experience (QoE). The use of standardized QoE metrics enables *estimation* of end-user perceived quality *without involving human subjects*. By using QoE metrics rather than conducting user studies, we are able to assess quality in an extensive sensitivity study involving a broad range of buffer size and workload configurations. More precisely, we evaluate these metrics over a wide range of end-user applications, (i.e., Web browsing, VoIP, and RTP video streaming) and workloads in two realistic testbeds emulating access and backbone networks. Each application type is analyzed over Internet-like traffic scenarios—without isolation in separate QoS classes—and over a range of buffer sizes.

Our main observations are the following:

1. We mainly find *network workload*, rather than buffer size, to be the primary determinant of end-user QoE. As intuitively expected, sustainable congestion impacts both QoS and QoE metrics by keeping the queue of the bottleneck buffer filled. Large (bloated) buffers amplify this effect. In the absence of congestion, however, (even bloated) buffer sizes impact QoS metrics, as observed

by previous studies, e.g., [83], but affect QoE metrics only marginally. The good news for network operators is that limiting congestion, e.g., via QoS or over-provisioning, can yield more immediate QoE improvements than efforts to optimize buffering.

2. We show the perceptual (QoE) perspective on buffering to differ from the known QoS perspective. This further emphasizes the use of application specific and perceptual metrics in Internet measurements. In this regard, the insights gained in this chapter serve as an example of the use of QoE metrics for measurement studies.

The remainder of this chapter is organized as follows. In Section §6.1, we review buffer sizing schemes. We present a passive measurement analysis on the prevalence of *buffer bloat* in Section §6.2. While in Section §6.3 we present our methodology to investigate QoE degradations, we discuss our experimental results in Section §6.4. Finally, we summarize our results in Section §6.5.

6.1 Background

The rule-of-thumb [181, 293] for dimensioning network buffers relies on the bandwidth-delay-product (BDP) $RTT * C$ formula, where RTT is the round-trip-time and C is the (bottleneck) link capacity. The reasoning is that, in the presence of *few* TCP flows, this ensures that the bottleneck link remains saturated even under packet loss. This is not necessary for links with a large number of concurrent TCP flows (e.g., backbone links). It was suggested in [293] and convincingly shown in [73, 83] that much smaller buffers suffice to achieve high link utilizations. The proposal is to reduce buffer sizes by a factor of $\sqrt{n}$ as compared to the BDP, where n is the number of concurrent TCP flows [73]. Much smaller buffer sizes have been proposed, e.g., drop-tail buffers with $\approx 20 - 50$ packets for core routers [138]. However, these come at the expense of reduced link utilization [83]. This problem has been addressed by a modified TCP congestion control control scheme that aims to maintain high link utilizations in small buffer regimes [161]. For an overview of existing buffer sizing schemes we refer the reader to [294].

While the above discussion focuses on backbone networks, more recent studies focus on access networks, e.g., [130, 196, 218, 284], end-hosts [10], and 3G networks [169]. These studies find that excessive buffering in the access network exists and can cause excessive delays (e.g., on the order of seconds). This has fueled the recent bufferbloat debate [154, 302] regarding a potential degradation in Quality of Service (QoS). Indeed, prior work has shown that buffer sizing impact QoS metrics. Examples include *network-centric* aspects such as per-flow throughput [246], flow-completion times [205], link utilizations [83], packet loss rates [83], and fairness [296]. Sommers *et al.* studied buffer sizing from an operational perspective by addressing their impact on service level agreements [274]. However, QoS metrics and even SLAs do not necessarily reflect the actual implications for the end-user. A first step towards investigating the impact of buffering on gaming QoE has been made in simulations for Poisson traffic [266]. In the remainder of this chapter, we present a *QoE centric* study that broadly investigates the impact of buffering and background traffic by using realistic testbed hardware and Internet like traffic scenarios.

6.2 Buffering in the wild

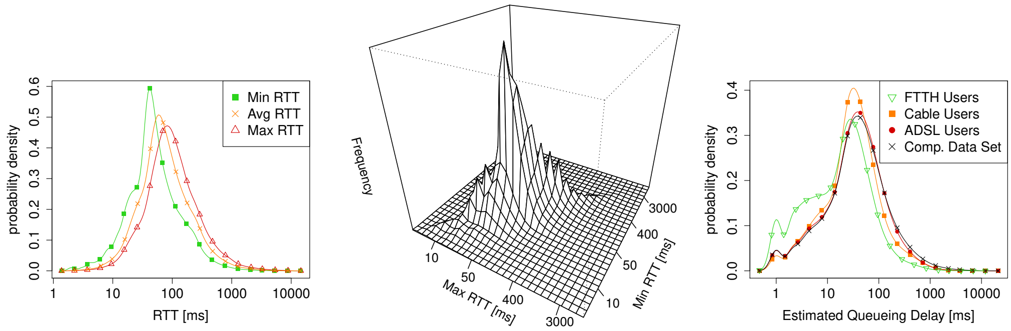
Before investigating the *impact* of buffering on QoE, we first motivate our study by investigating the *occurrence* of buffering in the wild. Our analysis is based on snapshots of Linux kernel level TCP statistics for 430 million randomly selected TCP/HTTP connections captured at a major Content Distribution Network (CDN). The data was collected at different vantage points, located primarily in central Europe, over a period of five months in 2011. All flows were established by end-users to retrieve content from the respective CDN caches, thus they capture typical Web browsing activity. This data corpus represents a significant sample of Internet users. It includes 81 million unique IP addresses originating from 22,490 autonomous systems (roughly 60% of the total advertised ASes when capturing the trace), located in more than 220 countries. Due to the vantage point locations, 56% of the IPs are located in central Europe.

We build our evaluation on smoothed RTT (sRTT) information reported in the data set. Smoothed RTT values are estimated by the TCP stack using Karn's algorithm and are provided by the kernel level TCP statistics. For each TCP connection, the data set reports *(i)* the minimum sRTT, *(ii)* the average sRTT, *(iii)* the maximum sRTT, and *(iv)* the number of samples. To evaluate the variability due to queuing, we focus on flows that have at least 10 RTT samples. The distribution (PDF) of the logarithm of the minimum, average, and maximum RTT is shown in Figure 6.1(a). The plot highlights that the average and maximum RTT deviate significantly from the minimum RTT, which is one indicator of possible queuing. Figure 6.1(b) underlines this intuition by showing the relationship of minimum and maximum RTT per flow in a 2D histogram. The figure shows that the maximum RTT significantly differs from the minimum RTT per flow, which further suggests the presence of queuing.

We estimate the queuing delay by evaluating the sRTT range (i.e., max-min) for each connection with at least 10 RTT samples. The implicit assumption is that the minimum RTT accounts for an empty queue and that queuing is the only source of delay variations. In general, additional factors such as route changes and layer 2 delays—particularly prominent in wireless networks—also contribute to delay variations. Since we cannot distinguish these factors from queuing delays, our estimation overestimates queuing and thus yields an *upper bound* on the magnitude of queuing.

We show the PDF of the logarithm of the estimated queuing delay in Figure 6.1(c). Based on WHOIS and DNS information, we split the complete data set into ADSL, Cable, and FTTH users and show their respective queuing delay distribution. Using this scheme, we associate 70% the flows to ADSL users, 1.4% to Cable users, and 0.02% to FTTH users. Most of the user flows experience a modest amount of queuing; 80% of all the flows experience less than 100ms of delay variation. Only 2.8% (1%) experience excessive queuing delays of more than 500ms (1000ms). This corresponds to only 2.5% (2%) of the observed hosts. We also consider user proximity to the CDN caches. Specifically, we examine flows with minimum RTT $\leq$ 100ms. In this setting, even more flows experience modest amounts of queuing: 95% (99.9%) of all connections have a queuing delay of less than 100ms (1sec), respectively.

Recently, the issue of *buffer bloat* has attracted significant attention. The debate is based on observations (e.g., [196]) showing that bufferbloat *can* happen, rather than it *does* happen. Despite this lack of empirical evidence, the bufferbloat argument has been used to motivate engineering changes in Internet standards (e.g., [153]) and to motivate new AQM approaches (e.g., CoDeL [232]). Two very recent studies examined the magnitude of the problem based on data from 118K [69] and 25K hosts [115], respectively and concluded that the magnitude of bufferbloat is modest. Our results, based on a much large data set of 80M hosts that is representative for a significant body of Internet users, further substantiate these findings. We empirically study whether Internet users at large experience excessive delays and we conclude that these do indeed occur, but only for a small number of flows and hosts. Thus—despite what is often claimed by the bufferbloat community—our findings further confirm the modest magnitude of excessive queueing delays. One explanation is that uplink capacity in the access, where bufferbloat has been found, is seldom utilized.



(a) Min, Avg, and Max RTT Distribution

(b) Min vs. Max RTT per Flow

(c) Queueing Delays

Figure 6.1: Occurrence of queuing in the wild

Our study of buffering in the wild is the starting point for our evaluation of the impact of buffering on QoE, including the case of excessive buffering (bufferbloat). While we estimate the magnitude of bufferbloat to be modest, its implications on QoE are largely unknown. For instance, a single delayed flow can severely degrade the QoE of an entire HTTP transaction. To shed light on the QoE impact of buffering, we first briefly introduce QoE, and then conduct a multi-factorial testbed study covering a wide range of end-user applications, buffer configurations, and traffic scenarios.

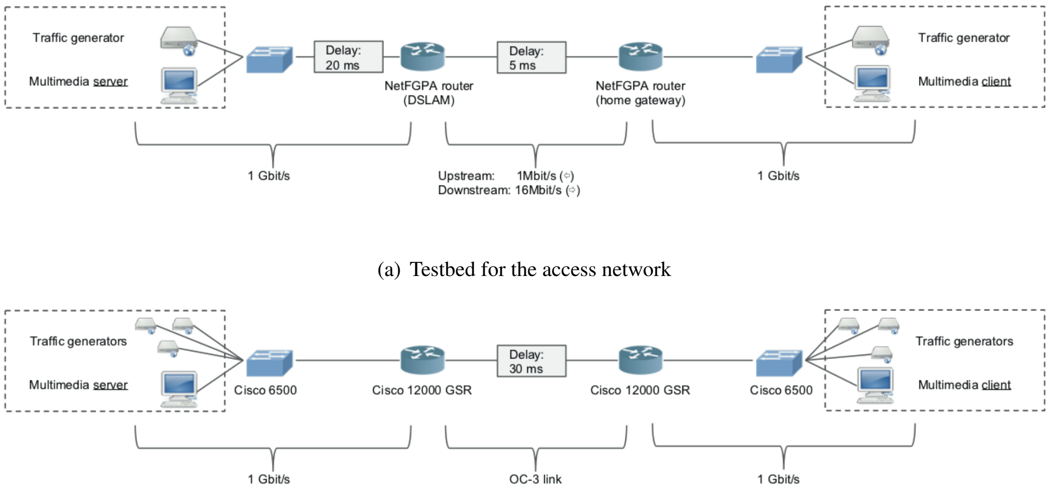
We use a testbed driven approach to study the impact of buffer sizes on the user perception (QoE) of common types of Internet applications: *i)* Voice over IP, *ii)* RTP/UDP video streaming as used in IPTV networks, and *iii)* Web browsing.

6.2.1 Testbed setup

We consider two scenarios: *i)* an access network and *ii)* a backbone or core network. Each scenario is realized in a dedicated testbed as shown in Figures 6.2(a) and 6.2(b). We use a testbed setup to have full control over all parameters including buffer sizes and generated workload.

As most flows typically experience only a single bottleneck link, both testbeds are organized as a dumbbell topology with a single bottleneck link, configurable buffer sizes, and a client and a server network. The hosts within the server (client) network on the left (right) side act as servers (clients), respectively. In the backbone case we configured the bandwidth and the delays of all links symmetrically. For the access network we use an asymmetric bottleneck link. In the backbone case we only consider data transfers from the servers to the clients. For the access network we also include data uploads by the clients—as they mainly triggered the bufferbloat debate [154].

The access network testbed, see Figure 6.2(a), consists of two Gigabit switches, four quad-core hosts equipped with 4 GB of RAM and multiple Gigabit Ethernet interfaces. Moreover, two hosts are equipped with a NetFPGA 1 Gb card each. The hosts are connected via their internal NICs to the switch to realize the client/server side network. The NetFPGA cards run the Stanford Reference Router software and are thus used to realize the bottleneck link. Thus the NetFPGA router and the multimedia hosts are located on the same physical host. As the NetFPGA card is able to operate independent of the host, it does not impose resource contention. The right NetFPGA router acts as the home router, aka DSL modem, whereas the left one acts as the DSLAM counterpart of the DSL access networks. To capture asymmetric bandwidth of DSL we use the hardware capabilities of the NetFPGA card to restrict the uplink and downlink capacities to approximately 1 respectively 16 Mbit/s. We use hardware to introduce



(a) Testbed for the access network

(b) Testbed for the backbone network

Figure 6.2: Testbed design for studying the impact of excessive buffering on QoE.

a 5 ms respectively 20 ms delay between the client (server) network and the routers. The 5 ms delay corresponds to DSL with 16 frame interleaving or to the delays typical for cable access networks [78]. The 20 ms account for access and backbone delays. While we acknowledge that delays to different servers vary according to a network path, a detailed study of path delay variation is beyond the scope of this dissertation. This is also the reason we decided to omit WiFi connectivity which adds its own variable delay characteristics due to layer-2 retransmissions. Instead, we focus on delay variations induced by buffering.

To be able to scale up the background traffic to the backbone network, see Figure 6.2(b), we include eight hosts, four clients and four servers. Each has again a quad-core CPU, 4 GB of RAM, and multiple Gigabit Ethernet network interfaces. The client/server networks are connected via separate Gigabit switches, Cisco 6500s, to backbone grade Cisco 12000GSR routers. Instead of using 10 Gbit/s and soon to be 100 Gbit/s interfaces for the bottleneck link, we use an OC3 (155 Mb/s nominal) link. The reason for this is that we wanted to keep the scale of the experiments reasonable, this includes, e.g., the TCPDump files of traffic captures. Moreover, scaling down allows us to actually experience bufferbloat given the available memory within the router. We use multiple parallel links between the hosts, the switch, and the router so that it is possible for multiple packets to arrive within the same time instance at the router buffer. With regards to the delays we added a NetPath delay box with a constant one-way delay of 30 ms to the bottleneck link. 30 ms delay roughly corresponds to the one-way delay from the US east to the US west coast. We again note, that the path delays in the Internet are not constant. However, variable path delays are beyond the scope of this dissertation. Instead we focus on delay variability induced by buffering. Moreover, we eliminate most synchronization potential by our choice of workload (see §6.3.1). To gather statistics and to control the experiments we always use a separate Ethernet interface on the hosts as well as a separate physical network (not shown).

	Uplink		Downlink	
Buffer Size	Delay	Scheme	Delay	Scheme
8	98	≈ BDP	6	min
16	198		12	
32	395		24	
64	788		49	≈ BDP
128	1,583		97	
256	3,167	max	195	max

Table 6.1: Access-network testbed: Buffer size configurations(in packets) and corresponding maximum queuing delay (in ms)

Buffer Size	Delay	Scheme
8	0.6	≈ TinyBuf
28	2.2	Stanford
749	58	≈ BDP
7490	580	10 × BDP

Table 6.2: Backbone-network testbed: Buffer size configurations(in packets) and corresponding maximum queuing delay (in ms)

6.3 Methodology: Testbed-driven measurements

6.3.1 Traffic scenarios

We use the Harpoon flow level network traffic generator [275] to create a number of congestion scenarios which range from no background traffic (noBG) to fully overloading (short-overload) the bottleneck link. Congestion causes packets from both the background traffic as well as the application under study to be queued or dropped just before the bottleneck link. Depending on the fill grade of the buffer, the size of the buffer, and the link speed, this will increase the RTT accordingly (see Tables 6.1 and 6.2). Overall, we use 12 scenarios for the access testbed and 6 for the backbone. We consider more for the access to distinguish on which links (i.e., upstream, downstream, or both) the congestion is subjected to.

In terms of the traffic that imposes the congestion we distinguish two different kinds of scenarios (see Tables 6.3 and 6.4): (*i*) long-lived TCP flows (long) and (*ii*) long-tailed file sizes to be able to resemble self-similar traffic as seen in today's networks (e.g., in core networks). For the latter, we choose Weibull distributed file sizes with a shape of 0.35 as their mean and standard deviation are finite as opposed to those of the often used Pareto distributions with a shape > 2. The generated traffic results in a mixture of bursty short-lived and long-lived flows with a mean of 50 KB. As the number of short flows dominates the number of long flows we refer to these scenarios as "short".

For scenarios with long-lived flows (long) we use flows of infinite duration. In this case the link utilization is almost independent of the number of concurrent flows. For long-tailed file sizes the workload of each scenario is controlled via the number of concurrent sessions that Harpoon generates. A session in Harpoon is supposed to mimic the behavior of a user [275] with a specific interarrival time, a file size distribution, and other parameters. We used the default parameters except for the file size distribution. In addition, we rescaled the mean of the interarrival time for the access network, as Harpoon's default parameters are geared towards core networks with a larger number of concurrent flows. For the access network we distinguish between few and many concurrent flows which results in medium and high load for the downstream direction and high load for the upstream, see Table 6.5. To impose different levels of congestion we adjusted the number of sessions for the backbone scenario to result in low, medium, high, and overload scenarios which correspond to link utilizations as shown in Table 6.6.

We checked that all hosts are using a TCP variant with window scaling. Due to the Linux version used the background traffic uses TCP-Reno in the backbone and TCP BIC/TCP CUBIC for the access. However, note that this does not substantially impact the QoE results as the applications VoIP and video

Scenario	Workload type	Flow Interarrival Distribution	File Size Distribution	# Sessions Up	# Sessions Down
noBG	No bg. traffic	–	–	–	–
	Upstream			1	–
short-few	Bidirectional	$\exp(\lambda = \frac{1}{2})$	$\text{weibull}(k = 0.35, \lambda = 10039)$	1	8
	Downstream			–	8
	Upstream			1	–
short-many	Bidirectional	$\exp(\lambda = \frac{1}{2})$	$\text{weibull}(k = 0.35, \lambda = 10039)$	1	16
	Downstream			–	16
	Upstream			1	–
long-few	Bidirectional	–	∞	1	8
	Downstream			–	8
	Upstream			8	–
long-many	Bidirectional	–	∞	8	64
	Downstream			–	64

Table 6.3: Workload configurations for the access network testbed

Scenario	Flow Interarrival Distribution	File Size Distribution	# Sessions Up	# Sessions Down	#flows
noBG	–	–	–	–	–
short-low	$\exp(\lambda = 1)$	$\text{weibull}(k = 0.35, \lambda = 10039)$	–	3*10	18
short-medium	$\exp(\lambda = 1)$	$\text{weibull}(k = 0.35, \lambda = 10039)$	–	3*30	49
short-high	$\exp(\lambda = 1)$	$\text{weibull}(k = 0.35, \lambda = 10039)$	–	3*60	206
short-overload	$\exp(\lambda = 1)$	$\text{weibull}(k = 0.35, \lambda = 10039)$	–	3*256	2170
long	$\exp(\lambda = 1)$	∞	–	3*256	675

Table 6.4: Workload configurations for the backbone network testbed

Scenario	Workload Type	Link Utilization [%] Mean Up	Link Utilization [%] Mean Down	Link Utilization [%] Sd Up	Link Utilization [%] Sd Down	Packet Loss [%] Up	Packet Loss [%] Down
noBG		–	–	–	–	–	–
	Upstream	98.9	0.3	0.7	0.1	34.7	0
short-few	Bidirectional	95	8.5	5.6	15.2	58.6	0.7
	Downstream	27.8	44.1	13.7	25.1	1.4	3
	Upstream	98.9	0.3	0.7	0.1	33.1	0
short-many	Bidirectional	93.3	10.7	4.3	20.1	60.9	1.3
	Downstream	53.8	78.7	12.8	23.5	4	4.5
	Upstream	99	0.2	0.7	0.1	33.1	0
long-few	Bidirectional	71.9	83.1	8.9	12.6	41.7	0.6
	Downstream	39.5	99.9	1.9	0.6	0.1	0.5
	Upstream	98.9	0.3	0.7	0	14.4	0
long-many	Bidirectional	83.8	61.8	11.2	26.4	60.7	0.2
	Downstream	68.5	99.6	3.9	4.9	0.03	9.3

Table 6.5: Traffic statistics for the access network testbed under workload using the BDP rule-of-thumb

Scenario	**Link Utilization [%]**				**Packet Loss [%]**	
	Mean		**Sd**			
	Up	**Down**	**Up**	**Down**	**Up**	**Down**
noBG	–	—	–	–	–	–
short-low	–	16.5	–	11.6	–	0
short-medium	–	49.5	–	18.8	–	0
short-high	–	98	–	6.5	–	0.2
short-overload	–	99.7	–	2.2	–	5.2
long	–	99.7	–	0.1	–	3.8

Table 6.6: Traffic statistics for the backbone network testbed under workload using the BDP rule-of-thumb

rely on UDP and the Web page is relatively small. Moreover, since the results are consistent it suggests that using a TCP variant optimized for high latency does not change the overall behavior even when the buffers are large.

6.3.2 Buffer configurations

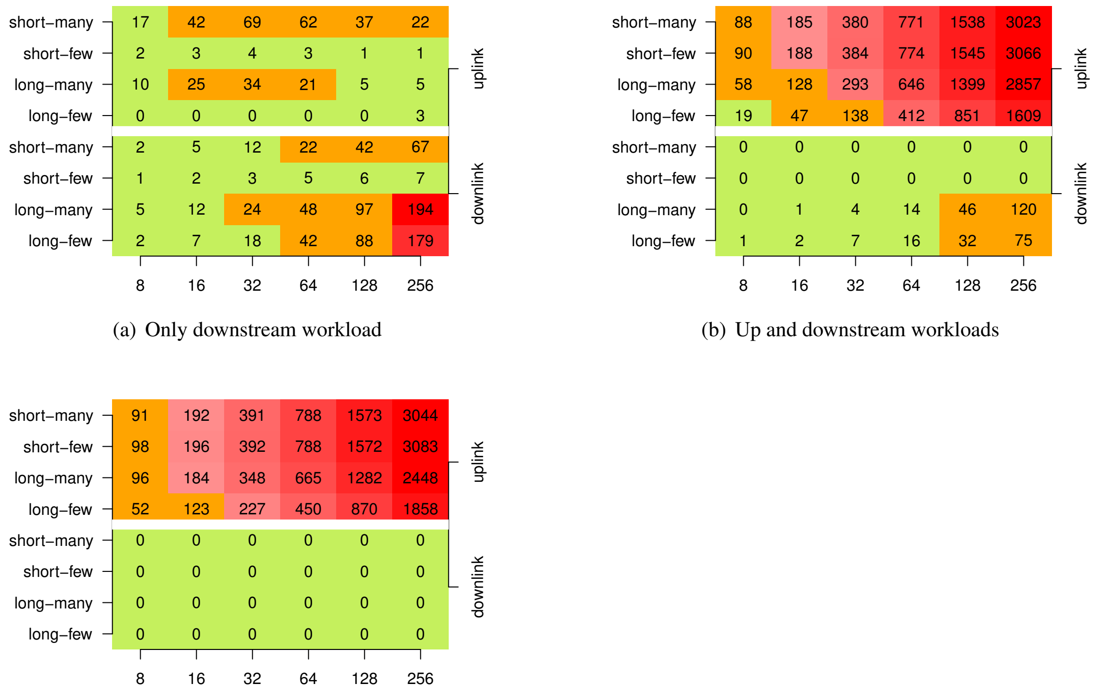
One key element of our QoE study is the buffer size configurations. Buffers are everywhere along the network path including at the end-hosts, the routers, and the switches. The most critical one is at the bottleneck interface, the only location where packet loss occurs. Therefore we focus on these and rely on default parameters for the others. For the bottleneck we choose a range of different buffer sizes, some reflect existing sizing recommendations, some are chosen to be small other large in order to capture extremes. Tables 6.1 and 6.2 summarize the buffer size configurations in terms of number of packets and shows the corresponding queuing delays.

For the access network we choose buffer sizes of powers of two, ranging from 8 to 256 packets. 256 is the maximum supported buffer size by the Stanford Reference Router software. For our choice of an asymmetric link (recall 1 Mbps uplink/16 Mbps downlink) the bandwidth-delay product (BDP) corresponds to roughly 8 and 64 packets, respectively. Since this set of buffer sizes yields delays up to buffer bloat, we consider the buffer configurations to approximate home router behavior.

For the backbone network we use *i)* the same minimum buffer size of 8 packets, which resembles the TinyBuffer scheme [138], depending on the largest congestion window achieved by the workloads. In addition, we use *ii)* 749 full-sized packets which corresponds to the BDP formula given an RTT of 60 ms, *iii)* 28 packet which corresponds to the Stanford scheme [73], i.e., BDP/$\sqrt{n}$, where $n = 3 * 256$ is the maximum number of concurrent for short-low, short-medium, short-high, and long (see Table 6.4), and *iv)* 10 × BDP packets an excessive buffering scheme.

6.4 Experimental results

After having designed the testbeds and corresponding workload and buffer configurations, we proceed to study the impact of excessive buffering on QoE. In this section, we first provide a brief QoS study, and continue with three different applications: Web, VoIP, and RTP video-streaming.



(a) Only downstream workload

(b) Up and downstream workloads

(c) Only upstream workload

Figure 6.3: Mean queuing delay (in ms) for the access networks testbed with different buffer size (x-axis) and workload (y-axis) configurations. Delays that significantly degrade the QoE of interactive applications (ITU-T Rec. G.114) are colored in red.

6.4.1 QoS

To highlight the potential importance of the buffer configuration on latencies, network utilization, and packet loss—the typical QoS values—we start our study with a detailed look at the background traffic. While the story is relatively straight forward for the backbone scenario, it is more complicated for the access network as the number of concurrent flows is smaller and there are subtle interactions between upstream and downstream.

To illustrate how the workloads and buffer sizes effect real-time applications, we conducted experiments to measure the latency introduced by the buffers. For this purpose we use the detailed buffer utilization statistics of the FPGA cards. Figure 6.3 shows the corresponding mean delays as heatmaps. We use three different heatmaps: one each for downstream/upstream workload only and one for combined up- and downstream workload. Each heatmap has two subareas—one for upstream at the top and one for downstream at the bottom. Each heatmap cell show the mean delay for a specific buffer size configuration and workload scenario, measured over two hours. The color of the heatmap cells correspond to categories of the ITU-T Recommendation G.114 which classifies delays based on their potential to degrade the QoE of interactive applications: green (light gray) is acceptable, orange (medium gray) problematic, and red (dark gray) causes problems.

In principle, we see that larger buffers sizes can increase the delays significantly independent of the workload. For the downlink direction the maximum delay is less than 200 ms. However, this can differ for the uplink direction. In particular, we observe delays of up to three seconds for larger–over-sized–

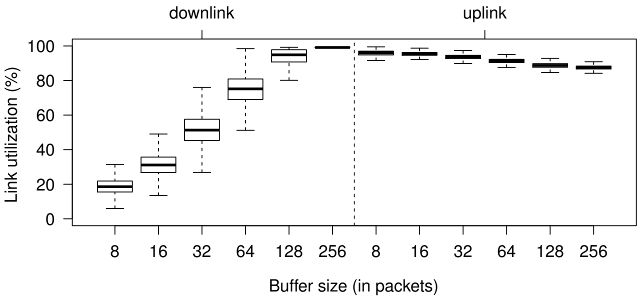


Figure 6.4: Link utilization for an asymmetric access link with various buffer sizes. The uplink and the downlink are simultaneously congested by 8 and 64 long-lived TCP flows, respectively.

buffers when the upstream is used for the uplink direction. This is almost independent of the workload. Overall, these delays are consistent with observations by Gettys [302] which started the bufferbloat discussion. Given these high latencies, we investigate the link utilization. Figure 6.4 shows a boxplot of the link utilization for the various buffer sizes in the scenario with simultaneously downloads and uploads (bidirectional workloads). The left/right half focuses on the downlink/uplink utilization. The uplink utilization is almost 100% while the downlink utilization ranges from 20% to 100%. Consistent with related work, we see that very small buffers can lead to underutilization while very large buffers can lead to large delays.

Comparing these link utilizations to those with no upstream workload (not shown) we find that, for bidirectional workloads, the buffer configurations below the BDP do not always fully utilize the downlink direction. Buffer sizes that correspond to the BDP yield full downlink utilization in the absence of upload workload, but not with concurrent download and upload activities. This phenomena can be explained by the queuing delay introduced by bloated uplink buffers that *virtually* increase the BDP thus rendering the downlink under-buffered. Related work coined the problem of bidirectional TCP flows that influence each other *data pendulum* [172]. In contrast to related work, our analysis highlights interdependencies between buffers and suggests that buffers should not be sized independent of each other. The phenomenon of low link utilization can be mitigated by counter-intuitively “bloating” the downlink buffer. Considering the delays observed in Figure 6.3(b), the BDP increases beyond the initial buffer size of 64 to 835 full sized packets. Note, that we can get full link utilization for buffers of smaller than 835 packets as we have a sufficient number of concurrent flows active.

In summary, the latency introduced by the buffers in home routers, aka, the uplink, might not only i) harm real-time traffic applications (due to excessive buffering), but also ii) drastically reduce TCP performance (due to insufficient buffering) in case of bidirectional workloads in asymmetric links. In effect it invalidates the buffer dimensioning assumptions due to the increase in RTT.

6.4.2 Web QoE

We next move to Web browsing, our first application under study. The Web browsing experience (WebQoE) can be quantified by two main indicators [135]. One is the *page loading time* (PLT), which is defined as the difference between a Web page request time and the completion time of rendering the Web page in a browser. Another one is the time for the first visual sign of progress. In this chapter we consider PLT of information retrieval tasks, for which there exists an ITU QoE model (i.e., G.1030 [45]) to map page loading times to user scores.

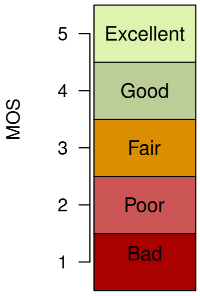


Figure 6.5: MOS scale for video and Web

We note that WebQoE does not directly depend on packet loss artifacts, but rather on the completion time of underlying TCP flows. Thus, factoring in various workloads and buffer sizing configurations—which influence the TCP performance—is particularly relevant for understanding WebQoE from a network only perspective. Given that the PLT as measured in a browser can be approximated from flow completion times as parameter, is sometimes considered as a QoS parameter. Since the applied G.1030 model logarithmically maps PLT to QoE, it can be misbelieved QoS parameters can (always) be mapped to QoE. We therefore note that other QoE models are of higher complexity as different input parameter are used that cannot be directly derived from a QoS parameters, e.g., speech signals as used in Section 6.4.3.

Approach

To evaluate the WebQoE, we map the PLT to a user score z by using the ITU Recommendation G.1030 [45] specified for Web information retrieval tasks. We consider the one-page version of the ITU model, which logarithmically maps *single* PLT's to scores in the range $z \in [1, 5]$ (i.e., 5: *excellent*, 4: *good*, 3: *fair*, 2: *poor*, 1: *bad*, as shown in Figure 6.5. This mapping uses six seconds as the maximum PLT, i.e., mapping to a "bad" QoE score. The minimum PLT—mapping to "excellent"—is set to 0.56 (0.85) seconds for access (backbone) scenario, due to different RTTs.

We remark that WebQoE research has advanced beyond factors captured in the applied ITU G.1030 [45] model. This concerns the impact of distraction factors such as noise or traffic on quality perception [166], task and content dependent factors [279], or task completion times and loading pattern [280]. These advances have, however, not yet converged to a revised model that is applicable in this study. Beyond additional factors, WebQoE research addresses the need for interactive Web use that goes beyond information retrieval tasks which follow request response pattern [136]. We remark that no QoE models fully addresses interactive Web usage (such as AJAX requests), which is why we must leave this aspect for future work. For the information retrieval scenario considered in this section, however, recent findings suggest the logarithmic dependency of waiting time and QoE [136, 280]—as used in applied G.1030 model—to remain valid. We thus stick to using ITU Recommendation G.1030 [45] as the current standard for assessing WebQoE.

To measure the PLT's, we consider a single static Web page, located in one of the testbed servers, and consisting of: one html file, one CSS file, and two medium JPEG images (sized to 15, 5.8, 30, and 30 KB, respectively). The Web page is loaded within 14 RTTs, including the TCP connection setup and teardown. Choosing a relatively small Web page size was inspired by the frequently accessed Google front page designed to quickly load. To retrieve this Web page we use the *wget* tool which measures

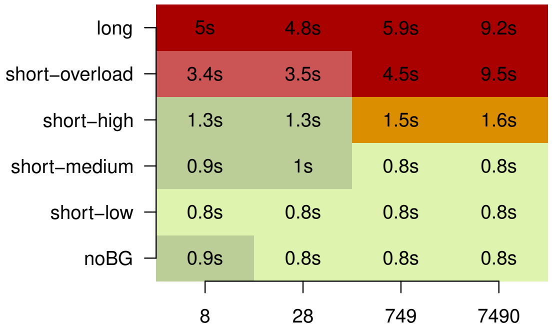


Figure 6.6: WebQoE Backbone: Median MOS (color) and page loading times (text) with different buffer size (x-axis) and workloads (y-axis).

the transfer time. *wget* is configured to sequentially fetch the Web page and all of its objects in a single persistent HTTP/1.0 TCP connection without pipelining. We point out that, as static Web pages have constant rendering times, it suffices to rely on *wget* rather than on a specific Web browser.

To further analyze the page retrieval performance, we rely on full packet traces capturing the HTTP transactions. We analyze the loss process of the captured TCP flows using the *tcpcsm* tool estimating retransmission events. We further measure the RTT during each experiment. We denote PLTs as RTT dominated if a significant portion of the PLT consists of the RTT component expressed by $14 * RTT$. Similarly, we denote PLTs as loss dominated if the increase in PLT can be mainly attributed to TCP retransmissions.

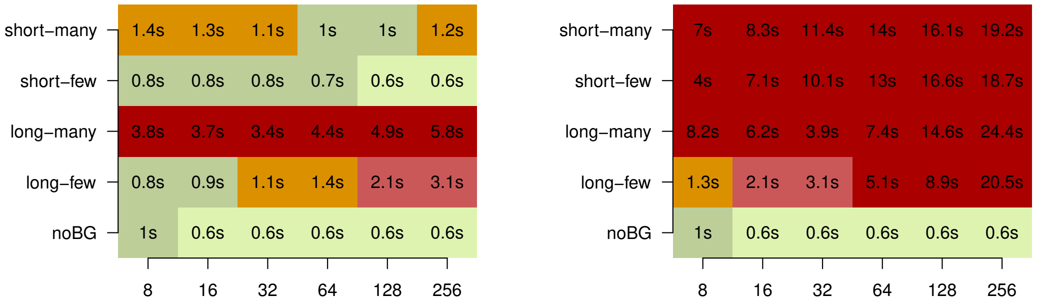
Backbone networks results

The median PLT and the corresponding QoE scores in the backbone setup are shown as a heatmap in Figure 6.6. As in the access scenario, the heatmap shows buffer sizes on the x-axis and the workload configuration on the y-axis. Each cell is colored according to the MOS scale from Figure 6.5 and displays the median PLT of 500 Web page retrievals.

The baseline results (noBG) show median page loading times of ≈ 0.8 seconds. These loading times are mainly modulated by $14 \times$ RTT (RTT $= 60$ ms, see § 6.2.1) needed to fully load the page (RTT component), making them higher than in the access network scenarios that has lower RTTs. In this scenario, the distribution of page loading times generally yields a slightly better performance for buffer sizes greater than or equal to the BDP; for these buffer configurations Web pages load up to 200 ms faster (80^{th} percentile not shown in the figure). The short-low scenario yields similar results despite the existence of background traffic.

We observe the first PLT degradations in the short-medium scenario for the 8 and 28 packets buffer configurations. In these cases, PLTs are affected by packet losses causing TCP retransmissions, while the 749 (BDP) and 7490 packet buffers absorb bursts and prevent retransmissions. As in the previous case, Web pages load up to 200 ms faster (80^{th} percentile not shown in the figure). The degradations in PLT are, however, small and only marginally affect the QoE score.

Degradations in the short-high scenario are twofold; while packet losses mainly affect the QoE for the 8 and 28 packets buffers, queuing delays degrade the QoE for the larger buffers. This effect is more pronounced in the short-overload and long scenarios that impose a higher link load. In these scenarios, the degradations for the 8 and 28 buffers are mainly caused by packet losses. The 749 and especially



(a) Access: only download activity

(b) Access: only upload activity

Figure 6.7: WebQoE Access: Median MOS (color) and page loading times (text) with different buffer size (x-axis) and workloads (y-axis).

the large 7490 buffer affected flow by introducing significant queuing delays; while the RTT doubles for the 749 buffer configuration, it increases by a factor of 10 for the 7490 buffer. Comparing short-overload to long for the 8, 28 and 749 buffer size yields a higher number of retransmissions in the long scenario, degrading the PLT. Concerning the PLT, short buffers of 8 and 28 packets show faster PLT for the short-high, short-overload, and long scenarios. However, improvements in the PLT do not help to generally improve the QoE as the PLTs are already high, causing bad QoE scores.

Our findings highlight the trade-off between packet loss and queuing delays. While larger buffers prevent packet losses and therefore improve the PLT in cases of less utilized queues/links, the introduced queuing delays degrade the performance in scenarios of high buffer/link utilization. In the latter, shorter buffers improve the PLT by avoiding large queuing delays, despite the introduced packet losses. The "right" choice in buffer size therefore depends on the utilization of the link and the buffer.

Access network results

Figures 6.7(a) and 6.7(b) show heatmaps of the median Web browsing quality (MOS) for the access network. Each cell in the heatmap shows the median PLT of 300 Web page retrievals per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to Figure 6.5.

The baseline results, namely the ones without background traffic, are shown in the bottom row of each heatmap part, labeled noBG. The fastest PLT that can be achieved in this testbed is ≈ 0.56s. As all of the cells are green (light gray), we can conclude that in principle each scenario almost supports excellent browsing quality and that any impairment is due to congestion. In this respect, it turns out that, even without background traffic, the WebQoE can be degraded by (too) small buffers, e.g., 8 packets. Due to packet losses causing retransmissions, the PLT is increased to 1 second thereby changing the user perceived quality.

Download activity. Figure 6.7(a) focuses on the scenarios when there is congestion on the downlink. For the short-few scenario the downlink is not fully utilized, thus most scores do not deviate much from the baseline results. With this type of moderate workload browsing can benefit from the capacity of large buffers to absorb transient bursts and reduce packet losses. For instance, configuring the buffers size to 256 packets reduces the PLTs to the baseline results (as opposed to PLTs of 0.8s for the smallest buffer configuration). Likewise, for the short-many scenario, which involves more competing flows and imposes a higher link utilization, big buffers generally reduce PLTs. As the queueing delays for these scenarios are not excessive, i.e., they are bounded by 192 ms, see Tables 6.1 and 6.2, large buffers do in fact improve the QoE by limiting the loss rate. Bufferbloat is visible for the long-few scenario, where the median PLT increases with the buffer size, as the PLT is dominated by RTTs caused by large queuing delays. As for the previous scenario, the effects of various buffer sizes are clearly perceived by the end-user (yet in a different manner).

In contrast, the buffer size does not change the WebQoE in the long-many scenario. The larger number of competing flows reduces the per-flow capacity and lets the PLT increases beyond the users' acceptance threshold. Therefore, the perceived QoE, in contrast to the previous configuration, can not be improved by adjusting the buffer size. Nevertheless, from a QoS perspective, configuring an appropriate buffer size can let Web pages to load 2 seconds faster. This is not as straightforward since it involves considering the trade-off between small buffers (packet losses) and large buffers (combined effect of packet losses and large RTTs).

Upload activity. Figure 6.7(b) focuses on the scenarios when there is congestion on the uplink. As expected, congesting the uplink seriously degrades the link overall performance and thereby also the WebQoE. The perceived quality is degraded to the minimum for every buffer size configuration of the scenarios short-many, short-few, and the long-many. The only scenario where the browsing experience is slightly more acceptable is the long-few scenario if buffers are small. Such configuration reduces the median PLT from 20 to 1.3 seconds, which maps to a *fair* quality rating. From a QoS perspective, the figure shows that the PLT and the buffer size are strongly correlated to the QoE. A wise decision on the dimensioning of the buffers can reduce the PLT from 24.4 to 3.8 seconds (long-many). However, and in line with the previous observations, such reductions do not generally suffice to change the user perceived (*bad*) quality.

Combined upload and download activity. In the case of workloads in both, the uplink and downlink direction (not shown), the QoE is dominated by the upload activity. However, due to lower *overall* link utilization and shorter queueing delays (see §6.4.1), the median PLT are less than for the scenarios involving only uploads. The resulting scores generally map to *bad* quality scores; only the long-few workload shows better QoE for buffers ≤ 128 packets.

Key findings

Our observations fall into two categories: *i)* When the link is low to moderately loaded, larger buffers (e.g., BDP or higher) help minimizing the number of retransmissions that prolong the page transfer time and thus degrade WebQoE. *ii)* When the link utilization is high, however, this increases RTT and thus the page transfers become RTT dominated. Also, loss recovery times increase. Therefore, smaller buffers yield better WebQoE despite a larger number of losses. However, the impact of the buffer size on the QoE metric page loading time is ultimately marginal, although the QoS metric page loading time sees significant improvements. While this may seem weird at first, let us consider a twofold improvement of the page loading time from 9 seconds to 5 seconds. This improvement is large for the QoS metric, but it is insignificant for the QoE metric, as both 9 and 5 seconds map to "bad" QoE scores regardless the QoS performance.

6.4.3 Voice over IP (VoIP)

In IP networks speech signals can be impaired by QoS parameters (e.g., packet loss, jitter, and/or delay), talker echo, codec and audio hardware related parameters, etc. Regarding QoS parameters, packet losses directly degrade speech quality as long as forward error correction is not used as is typical today. Network jitter can result in losses at the application layer as the data arrives after its scheduled play-out time. Moreover, excessive delays impairs any bidirectional conversation as it changes the conversational dynamics in turn taking behavior.

Approach

We use a set of 20 speech samples recommended by the ITU [44] for speech quality assessment. Each sample is an error-free recording of a male or female Dutch speaker, encoded with G.711.a (PCMA) narrow-band audio codec, and lasts for eight seconds. Each of the 20 samples is automatically streamed, using the PjSIP library, over our two evaluation testbeds, see § 6.3 and subjected to the various workloads. PjSIP uses the typical protocol combination of SIP and RTP for VoIP. We remark that we do not consider other situational factors such as the users' expectation (e.g., free vs. paid call) [224] which can also affect the perceived speech quality (see § 2.3.2). For the VoIP QoE assessment, we separately evaluate speech signal degradations and conversational dynamics, using two widely used and standardized QoE models: PESQ and E-Model. Individual scores are combined to the final QoE score.

Speech signal degradations. To assess the speech quality of each received output audio signal, relative to the error-free sample signal, we use the Perceptual Speech Quality Measure (PESQ) [42] as standardized model. PESQ takes as input both the error-free audio signal and the perturbed audio signal, and computes the QoE score z_1. Note that while z_1 is *influenced* by loss and jitter, the QoE estimation is *signal based* and *not a function of QoS parameters*. The influence of loss and jitter on z_1 can therefore not be quantified.

Conversational dynamics. The PESQ model only accounts for the perceived quality when listening to a remote speaker but does not account for conversational dynamics, e.g., for humans taking turns and/or interrupting each other. This can be impaired by excessive delays and thus can degrade the quality of the conversation significantly [188, 224, 250, 263]. Thus, according to the ITU-T recommendation G.114 one-way delays should be below 150 ms (or at most 400 ms). Therefore, we measure the packet delay during the VoIP calls. We now use the delay impairment factor of the ITU-T E-Model [43] to get a score z_2. We remark that even though z_2 is computed using a standardized and widely used model, it is subject to an intense debate within the QoE literature as there is a dispute about the impact of delay on speech perception [137, 188, 250]. Among the reasons is that the delay impact depends on the nature of the conversational task (e.g, reading random numbers vs. free conversation) as well as the level of interactivity required by the task [188]. Thus, there can be mismatches between the quality ratings of the E-Model and tests conducted with subjects.

Overall score. The range of the score z_1, which captures loss and jitter, is $[1, 5]$. We remap it to $[0, 100]$ according to [281]. The range of the score z_2, capturing the delay impairment, is $[0, 100]$. Note, the semantics of z_1 and z_2 are reversed: a large value for z_1 reflects an excellent quality; however, a large value for z_2 reflects a bad quality, and vice-versa. We combine the two scores to an overall one as follows: $z = \max\{0, z_1 - z_2\}$. Thus, if z_1 is good (i.e., due to negligible loss and jitter), but the z_2 is bad (i.e., due to large delays), then the overall score z is low, reflecting a poor quality and vice-versa. Finally, we map z to the MOS scale $[1, 5]$ according to the ITU-T recommendation P.862.2, see Figure 6.8; in the end, low values correspond to bad quality and high values to excellent quality.

Backbone networks results

Similar to the access network scenario, we show the voice quality in the backbone network scenario as a heatmap in Figure 6.9. The heatmap shows the median MOS for unidirectional audio from the left to the right side of the topology per buffer size (x-axis) and workload scenario (y-axis). Each cell in the

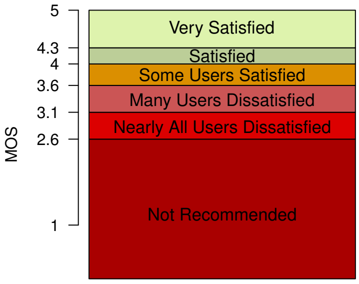


Figure 6.8: MOS scale for G.711

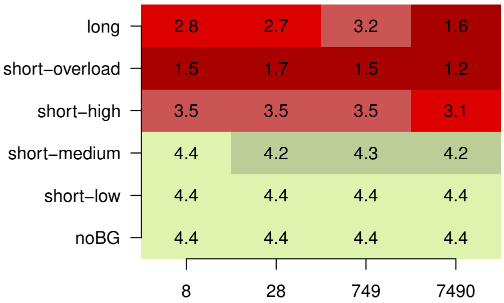


Figure 6.9: VoIP Backbone: Median Mean Opinion Scores (MOS) for voice calls with different buffer size (x-axis) and workload (y-axis) configurations.

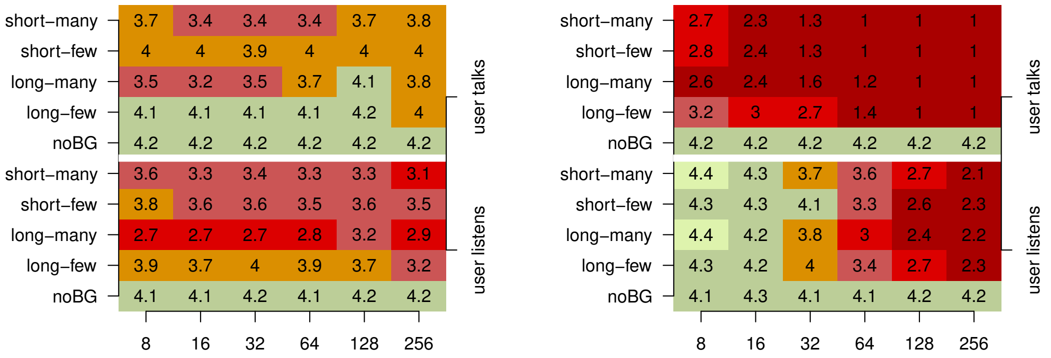
heatmap is based on 2000 VoIP calls. Here, each speech sample is send 100 times which is possible as the total number of scenarios is smaller. Notice, the bottom row label noBG shows the baseline results for an idle backbone without background traffic.

While the effects of the buffer size are less pronounced, the nature of the background traffic (long vs. short-*) and the link utilization (short-low to short-overload) are more significant. The type of workload can drastically degrade the quality score. Concretely, low to medium utilization levels as imposed by the scenarios short-low and short-medium, respectively, are close to the baseline results. In contrast, more demanding workloads such as the scenarios short-high and long, leading to higher link utilizations, and result in more than 1 point reductions in the MOS scale. Further, the aggressiveness of the workload further decrease the quality; the median MOS for the short-overload workload is 1.5 and thus significantly lower than for short-high and long that also lead to high link utilizations.

In general, the quality scores are, on a per workload basis, fairly stable across buffer-configurations below the BDP (749 packets). In these cases, we only observe small degradation of 0.4 points for the scenario long workload for the smallest buffer configuration. However, buffer configurations larger than the BDP, i.e, 7490 packets, lead to excessive queuing delays, which in turn lead to significant quality degradations of the z_2 delay impairment component. For example, the scores corresponding to the scenarios long and short-overload workloads have MOS values of almost half of their counterpart with the BDP configuration.

Access networks results

Figures 6.10(a) and 6.10(b) show heatmaps of the median call quality (MOS) for the access networks. Each cell in the heatmap shows the median MOS of 200 VoIP calls (each speech sample is send 10



(a) Access: download activity

(b) Access: upload activity

Figure 6.10: VoIP Access: Median Mean Opinion Scores (MOS) for voice calls with different buffer size (x-axis) and workload (y-axis) configurations. The heatmaps for the access networks include inbound calls (user listens) and outbound calls (user talks).

times) per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to the color scheme of Figure 6.8. The heatmap is divided into two parts (i) when user talks (upper part) and (ii) when the user listens to the remote speaker (bottom part).

The baseline results, namely the ones without background traffic are shown in the bottom row of each heatmap part, labeled noBG. They reflect the achievable call quality of the scenarios. As all of them are green, we can conclude that in principle each scenario supports excellent speech quality and that any impairment is due to congestion and not due to the buffer size configuration per se.

Download activity. Figure 6.10(a) focuses on the scenarios when there is congestion in the downlink. As there is no explicit workload in the uplink, one may expect that only the “user listens” part is effected but not the “user talks” part. This is only partially true as the “user talks” part of the heatmap shows deviations of up to 0.8 MOS points from the baseline score. These degradations are explained by the substantial number of TCP `ACK` packets, reflected by higher link utilizations (not shown). Recall, the uplink capacity is 1/16th only of the downlink capacity.

The degradations in “user listens” part of the heatmap are, as expected, more pronounced then for the “user talks” part. However, there are also significant differences according to the workload and the buffer configurations. For instance, with buffers sizes of 64 packets the long-many workload yields a median MOS of 2.8, whereas the long-few workload yields a median MOS of 3.5. Interestingly, even though the short-few workload does not fully utilize the downlink, i.e., less than 50% (not shown), it gets scores worse than a workload with higher link utilization, e.g., long-few. This is due to the higher jitter that is imposed by the large changes in link utilization and thus in the buffer utilization. With regards to buffer sizes we in general observe the worst scores for the larger buffer configurations, i.e., 256 packets due to the added delays. However, the best scores only deviate by 0.7 MOS points from this worst score (e.g., for the 8 packets buffer), suggesting that smaller buffers do not significantly improve audio quality. We conclude that the level and kind of workload has a more significant effect than the size of the buffer.

Upload activity. Figure 6.10(b) focuses on the scenarios when there is congestion on the uplink. According to the above reasoning one would therefore only expect degradations for the “user talks” part. This is not the case. The MOS in the “user listens” part of the heatmap decreases by 0.5 to 2 from the baseline results for all scenarios with buffer sizes ≥ 64. The reason for this is that the delays added by the excessive buffering in the uplink also degrade the overall score due to the delay impairment factor z_2. Since this factor expresses the conversational quality, it does not only effect the “user talks” but also the “user listen” part sent over the (non-congested) downlink. Excessive delays added by the buffers also explain the significant degradation of MOS values from 4.2 to $1 - 1.4$ for the “user talks” part. Due to the congestion, packet loss is also significant for all scenarios. This is why the best MOS value is limited to 3.2 even for short buffer configurations.

In the context of the bufferbloat discussion, Figure 6.10(b) corroborates that excessive buffering in the uplink yields indeed bad quality scores. Reducing the buffer sizes results in better MOS and contributes to mitigate the negative effects of the large delays introduced by the uplink buffer, e.g., the difference in the MOS for an inbound audio can be as high 2.5 points (long-many workload).

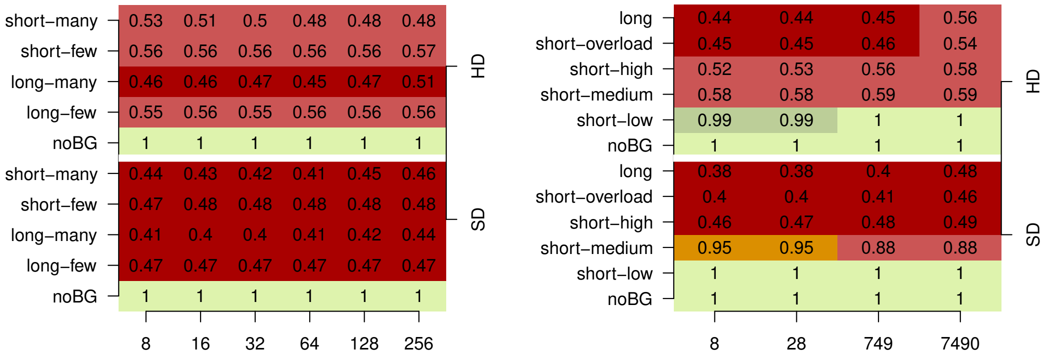
Combined upload and download activity. Scenarios (plot not shown) with upload and download congestion show similar results to scenarios with only uploads. Here as well the delays introduced by the uplink buffer dominate in both “user talks” and “user listens” parts. However, with combined upload and download activity, the “user listens” is slightly more degraded than with only upload activity. The reason for this is additional background traffic in the downlink that interacts with the voice call. For instance, with buffers configured to 16 packets, the long-few shows an additional degradation of 0.8 MOS points (thus mapping to a different rating scale). Limiting access congestion by isolating VoIP calls in a separate QoS class—as often implemented for ISP internal services but not Internet-wide—is therefore a good strategy.

Key findings

We find that VoIP QoE is substantially degraded when VoIP flows have to compete for resources in congested links. This is particularly highlighted in the backbone network scenario, where low to medium link utilizations yields good QoE and high link utilization ($> 98\%$) degrade the QoE. In the case of the latter, the congestion leads to insufficient bandwidth on the bottleneck link that affects the VoIP QoE. For access networks we show that, due to the asymmetric link capacities, the different audio directions can yield different QoE scores. For instance, in one direction (e.g., user talks) the speech quality might be acceptable, while it is impaired for the other (e.g., remote speaker talks) or vice-versa. Moreover, the speech quality is much more sensitive to congestion on the upstream direction than the downstream one. Due to the light queuing delays introduced by bloated buffers in the uplink, maintaining a conversation can be challenging in the presence of uplink congestion. For both access and backbone networks, configuring small buffers can results in better QoE. However, our results highlight that this may not suffice to yield “excellent” quality ratings. Thus, we advocate to use QoS mechanisms to isolate VoIP traffic from the other traffic. This is already common for ISP internal services but not for ISP external services.

6.4.4 Video-streaming with real-time protocol (RTP)

Next, we explore the quality of video streaming using the Real-time Transport Protocol (RTP) which is commonly used by IPTV service providers. RTP streaming can be impaired by packet loss, jitter, and/or delay. Again packet losses directly degrades the video as basic RTP-based video streaming *typically* does not involve any means of error recovery. Network jitter and delays result in similar impairments as with voice and include visual artifacts or jerky playback. However, they depend on the concrete error concealment strategy applied by the video decoder.



(a) Access: download activity

(b) Backbone

Figure 6.11: Median MOS (color) and SSIM (text) for HD and SD RTP video streams with different buffer size (x-axis) and workloads (y-axis).

Approach

We chose three different video clips from various genres as reference. Each video has a length 16 seconds. They are chosen to be representative of various different kinds of TV content and vary in level of detail and movement complexity. Thus, they result in different frame-level properties and encoding efficiency; *A)* an interview scene, *B)* a soccer match, and *C)* a movie. Each video is encoded using H.264 in SD (4 Mbps) as well as HD (8 Mbps) resolution. Each frame is encoded using 32 slices to keep errors localized. This choice of our encoding settings is motivated by our experiences with an operational IPTV network of a Tier-1 ISP. We use VLC to stream each clip using UDP/RTP and MPEG-2 Transport Streams. Without any adjustment VLC tries to transmit all packets belonging a frame immediately. This leads to traffic spikes exceeding the access network capacity. In effect VLC and other streaming software propagate the information bursts directly to the network layer. As our network capacity, in particular for the access, is limited we configured VLC to smooth the transmission rate over a larger time window as is typical for commercial IPTV vendors. More specifically, we decided to use a smoothing interval (1 second) that ensures that the available capacity is not exceeded in the absence of background traffic. The importance of smoothing the sending rate is often ignored in available video assessment tools such as EvalVid, making them inapplicable for this study.

We note that Set-top-Boxes in IPTV networks often use proprietary retransmission schemes that request lost packets once [174]. Due to the unavailability of exact implementation details we do not account for such recovery. Our results thus present a baseline in the expected quality; however, systems deploying active (retransmission) or passive (FEC) error recovery can achieve higher quality. We use two different full-reference metrics, PSNR and SSIM, to compute quality scores from the original and the perturbed video stream. While not considered as QoE metric, PSNR (Peak Signal Noise Ratio) enables a quality ranking of the same video content subject to different impairments [193, 289]. However, it does not necessarily correlate well with human-perception in general settings. SSIM (Structural SIMilarity) [300] has been shown to correlate better with human perception [301]. We map PSNR and SSIM scores to quality scores according to [309].

Backbone network results

Similar to the previous access network scenario, we show the video quality scores obtained for the same video C as a heatmap in Figure 6.11(b), both for SD and HD resolution. Each cell of the heatmap shows the median SSIM score and is colored according to the corresponding perceptive MOS score, see Figure 6.5. As in the previous scenario, the video was sent 50 times per buffer size (x-axis) and workload (y-axis) configuration. We omit PSNR quality scores as they are similar to the SSIM quality scores. As in the access network scenario, the bottom row labeled noBG shows the baseline results for an idle backbone without background traffic. Similarly, workloads that do not fully utilize the bottleneck link, i.e., short-low, lead to optimal video quality, as expressed by an SSIM score of 1. The reason is that the available capacity in the bottleneck link allows streaming the video without suffering from packet losses.

First quality degradations are observable in the short-medium scenario, where the quality decreases with increasing link utilization. In this scenario, workloads achieve full link utilization for 749/7490 buffers more often than for the 8/28 buffer configurations. It results in higher loss rates for the video flows lowering the quality and is more pronounced for the HD videos which have higher bandwidth requirements. Workloads that sustainably utilize the bottleneck link, i.e., short-high, short-overload, and long, yield bad quality scores due to high loss rates. These scenarios provide insufficient available bandwidth to stream the video without losses. Increasing the buffer size helps to decrease the loss rate, leading to slight improvements in the SSIM score. Comparing the obtained quality scores among the three different videos leads to minor differences in quality scores. These differences result from different encoding efficiencies that cause different levels of burstiness in the streamed video. However, the quality scores of all video clips lead to the same primary observation: quality mainly depends on the workload configuration and decreases with link utilization. Increasing the buffer size helps to lower the loss rate and therefore to marginally improve the video quality.

Access network results

We show our results as heatmap in Figure 6.11(a). The heatmap shows the quality score for video C sent 50 times per buffer size (x-axis) and workload (y-axis) combination. Each cell shows the median SSIM score and is colored according to the corresponding MOS score (see Figure 6.5); a SSIM score of 1 expresses excellent video quality, whereas 0 expresses bad quality. The upper and the bottom parts of the heatmap correspond to the results of HD and SD video streams, respectively. We omit quality scores obtained for the PSNR metric as they yield predicted scores similar to those obtained by SSIM. Also, as we focus on IPTV networks where the user consumes TV streams, no video traffic is present in the upstream. For this reason, we only show results for workloads congesting the downlink. To show the achievable quality for all buffer size configurations in the absence of background traffic, we show baseline results in rows labeled noBG. In these cases, the video quality is not degraded due to the absence of congestion.

In the presence of congestion, however, the SD video quality is severely degraded, expressed by a “bad” MOS score. This holds regardless of the workloads and the buffer configuration; the link utilization by all of the workloads cause video degradation due to packet loss in the video stream. We observe that even a low packet loss rate can yield low MOS estimates. Moreover, much higher loss rates (one order of magnitude bigger) can yield the same estimates. For instance, although both scenarios, long-few and long-many, have a similar SSIM and MOS score for buffers sized to 256 and 8 packets respectively, they show different packet loss rates of 0.5% and 12.5%.

In comparison to the SD video, degradations in HD videos are less pronounced although, in some cases, the packet loss rate is higher. For instance, the packet loss rate for HD and SD video streaming is, with the long-few workload and buffers sized to 256 packets, 2.6% and 1.3% respectively. However, the HD video stream obtains a better MOS score. This interesting phenomena can be explained by the higher resolution and bit-rate of HD video streams, which reduce the visual impact of artifacts resulting from packet losses during video streams. In the case of UDP video streaming in access networks, what matters is the available bandwidth, not the buffer size. Moreover, even though buffers regulate the trade-off between packet losses and delay, they have limited influence on the quality from the perspective of an IPTV viewer.

Key findings

Our results indicate a roughly binary behavior of video quality: *i)* when the bottleneck link has sufficient available capacity to stream the video, the video quality is good, and *ii)* otherwise the quality is bad. In between, if the background traffic utilizes the link only temporarily, the video quality is sometimes degraded. This results in an overall degradation that increases with link utilization. Using HD videos yields marginally better quality scores even though they use higher bandwidth. We find that the influence of the buffer size is marginal as delay does not play a major role for IPTV. What mainly matters is the available bandwidth. We did not include quality metrics relevant for interactive TV or video-calls. We further note that our results represent a baseline quality achievable without error recovery. Error recovery (e.g., retransmissions) will increase the overall quality.

6.5 Summary

The goal of our work is to elucidate the open problem of proper buffer sizing and to pave the way for more informed sizing decisions. In this respect, this chapter presents a comprehensive study of the impact of buffer sizes on *Quality of Experience*. By this we complement a large body of related work on buffering with a first look at factors relevant to end-user experience. This is a relevant view since it has implications for network operators and service providers, and by extension, device manufacturers.

To tackle this problem, we first evaluate the impact of buffering in the wild using a large data set from a major CDN that serves for a large number of Internet users (80M IPs from 235 countries). Our analysis shows that buffering is likely to be prevalent on a large scale. We also observe a rather modest amount of potential buffer bloat. This motivates our further evaluation of buffer sizing including the impact of very large buffers, i.e., buffer bloat.

The main contribution of this chapter is an extensive sensitivity study on the impact of buffer sizing on Quality of Experience. This is based on a testbed-driven approach to study three standard application classes (voice, video, and Web) in two realistic testbeds emulating access and backbone networks. Our evaluation considers a wide range of traffic scenarios and buffer size configurations, including bufferbloat.

Our main finding is that the *level of competing network workload* is the primary determinant of user QoE. It is generally known and understood that buffer sizing impacts QoS metrics. In particular, it is not surprising that sustainable congestion degrades network performance. Surprisingly, our results show that in the absence of congestion, buffer sizing has a significant impact on QoS metrics, whereas it only marginally impacts QoE metrics. The good news of this novel observation for network operators is that limiting congestion, e.g., via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering. There are, however, several subtle issues that complicate buffer sizing.

Concretely, application characteristics and the level of congestion determine the potential impact of buffer sizing choices. In the case of Web browsing, large buffers yield better QoE for moderate network loads, while smaller buffers improve QoE for high network loads. This suggests load-dependent buffer sizing schemes. Despite the potential for optimization, the impact of reasonable buffer sizes on QoE metrics is marginal, while the impact on QoS metrics can be significant. This is relevant for network operators, as it indicates that as long as buffers are kept to a reasonable size their impact is of marginal relevance. Concerning the ongoing bufferbloat debate, our main claim is that only relatively narrow conditions seriously degrade QoE, i.e., when buffers are over-sized and sustainably filled. Such conditions indeed occur in practice, as our empirical evaluation and other recent studies confirm, but their occurrence is relatively rare. We remark that emulations are by definition an abstraction of live networks and that predictive QoE models are abstractions of end-users. Thus our results should not be interpreted as representative of any specific network deployment or specific end-user quality ratings. We do, however, argue that our results accurately reflect the key interactions between buffer sizes and network traffic, which is the objective of our study.

Observed discrepancies among network-centric QoS metrics and application/user centric QoE metrics advocate a stronger use of application-centric metrics in measurement and performance evaluation studies. This is challenging since QoE is an application-specific measure and thus needs to be evaluated individually for every application. To reduce the complexity of QoE assessment in network design and network measurement, it appears appealing to aim for a general mapping of network performance (e.g., QoS metrics) to QoE. We believe this is possible for some QoE indicators, e.g., page-loading time in specific scenarios. However, since QoS and QoE represent fundamentally different concepts that depend on different parameters despite of common misconceptions QoE cannot be generally derived from QoS metrics. Examples used in this chapter include speech QoE assessment based on audio signals or video quality assessment based on decoded video frames. To simplify future QoE evaluations, this evaluation exemplifies the use of QoE metrics for measurement studies.

7 Conclusion and outlook

7.1 Summary

The World Wide Web has become the most used information system worldwide. It has fueled an unprecedented commercialization of the Internet by turning a system designed for academic data exchange into a widely used social medium. At the same time, the very same commercialization of the Web has created a complex ecosystem of infrastructure on the Internet on which the Web builds. This dissertation explores these two orthogonal aspects to complement the large body of literature that has focused on understanding the Web, its ecosystem, and its relation to the Internet (Chapter §2).

This dissertation posits that there are three enablers for today's Web: *i) content delivery*, which allows to scaling up and serving content to end users worldwide fast, *ii) content monetization*, which provides the economic resources needed to deploy such infrastructure and create content, and *iii) content search*, which allows users to find resources efficiently on the Web without the need to navigate from node to node. Consequently, in the first part of this dissertation we characterize Web traffic on the Internet concerning these three enablers (Chapters §3 and §4). To this end, we examine traffic at multiple vantage points on the Internet, including a residential broadband network, transit (long-haul) links of a tier-1 ISP, two European IXPs, and servers of a large CDN. Our analysis reveals two facets of Web traffic, for which we coin the terms *front-* and *back-office* Web traffic. The term *front-office* traffic refers to traffic exchanged between users and front-end servers. By contrast, the term *back-office* traffic refers to traffic exchanged between automated hosts acting as clients and other servers, e.g., communication between front-end servers and back-end or origin servers. This distinction within Web traffic is particularly relevant for our understanding of the Web's ecosystem. We argue that in the early days of the Internet, there was a strong resemblance between the end-to-end argument (see [258]) and the Web's application protocol (HTTP) client-server model: both Web end points hold most of the logic. However, this has faded away as shown by the increase of back-office Web traffic. Although the end-to-end argument continues to hold, service providers increase the amount of logic in their front-end servers: Retrieving a Web page or providing a Web service does not only involve a client fetching objects from multiple servers, but also these servers may contact other servers as intermediaries for the client. Consequently, we provide a study about *Web content delivery, monetization, and search* while differentiating between *front- and back-office* Web traffic.

In Chapter §3, we investigate *front-office* Web traffic, and more precisely, *content monetization* via the advertisements that are displayed to the end users. We devise a methodology to identify ad-related traffic as well as browsers using ad-blockers. We quantify the amount of ad-related traffic in one residential broadband network with 19.7K subscribers of a major European ISP (around 18% of the requests). We report the prevalence of ad-blockers at the same vantage point (22% of the most active users). Furthermore, we comment on configuration settings and effectiveness of the *acceptable ads* initiatives (see [3]). These observations are important in the context of *content monetization* because the current business model relies on the implicit agreement that users view ads in exchange for "free" content. We corroborate that not all users accept this *status-quo*, who perceive advertisements as intrusive and because they impair Quality of Experience. Moreover, we show how to infer *back-office* Web activity related to *content delivery and monetization* in *front-office* traffic.

In Chapter §4, we focus on *back-office* Web traffic. We devise a methodology that can identify this traffic on data sampled at the core infrastructure of the Internet. Specifically, we search for *back-office* Web traffic at two European IXPs with respectively 500 and 100 members as well as at two long-haul links of a Tier-1 ISP. Our results suggest that *back-office* Web traffic represents not only a significant fraction of today's Internet traffic but also today's Internet transactions. Its volume contribution ranges on average from 10% to 30% per vantage point and can even exceed 40% for some time periods. We further characterize it and put it into the context of the three fundamental functions aforementioned, i.e., *crawling* (to find and index Web content), *real-time bidding* (to make advertisements more effective), and *request forwarding* (mainly used by CDNs to scale with demand and improve performance). We find, for example, that at one of the IXPs auctioneers have a 22% share of the back-office requests but only 1% of the bytes, while crawlers contribute respectively roughly 10% and 15% to both. We corroborate our findings with an analysis conducted at one of the larger CDNs and characterize traffic that supports *content-delivery*. Our observations highlight the amount of complexity hidden to end users in support of *content delivery, monetization, and search*, and suggest that a significant amount of the Internet's infrastructure, traffic and interconnections is devoted to *back-office* Web traffic.

In the second part of this dissertation we investigate how choices concerning Internet infrastructure affect Web usage. As an *information system* that runs on the Internet, the Internet's infrastructure and its communication protocols can alter the way Internet users interact with the Web.

We investigate *Internet connectivity* in Chapter §5. Without it, users cannot reach Web content hosted on remote Web servers. One important issue regarding connectivity on the Internet is its transition from IPv4 to IPv6. This transition is necessary due to the scarcity of IPv4 addresses [253]. Such scarcity limits the deployment of new services and access networks on the Internet, and forces ISPs to explore technologies that can mitigate this problem. Consequently, two versions of the IP protocol coexist today on the Internet: IPv4 and IPv6. Related work shows that a large portion of the Web is not accessible over IPv6 and that there are still differences regarding performance among both protocols when browsing the Web. These two observations motivate our study on how the subscribers of dual-stack ISP use IPv4/IPv6 connectivity. We rely on passive measurements conducted at one of these ISPs, i.e., one residential broadband network with 12.9K subscribers. We design a methodology that allows us to characterize *IPv6 usage* and complement related work on *IPv6 adoption*. Namely, we investigate factors that hamper the growth of IPv6 traffic on such networks, including configurations of customer premises equipment (CPE), ISP connectivity, service availability (e.g., CDNs), and applications falling back to IPv4 due to poor IPv6 performance.

In fact, improving Web performance is one of the major goals among companies providing *content delivery* as a service, i.e., CDNs. We study in Chapter §6 one particular issue that affects Web browsing: large delays due to excessive buffering (*buffer bloat*). This phenomenon *may* occur when *transport* protocols like TCP interact with over-sized buffers. We use data from a major CDN that serves a large number of Internet users (80M IPs from 235 countries) to investigate when are buffers over-sized and sustainably filled. Such conditions indeed occur in practice, as our empirical evaluation and other recent studies confirm, but their occurrence is relatively rare. While the effects of buffering are understood in terms of QoS metrics, the impact on QoE is not. We tackle this by devising a methodology that allows us to *approximate* QoE degradations without the need of actual users. We perform a set of evaluations on two testbeds emulating *access* and *backbone* networks. We mainly find network workload, rather than buffer size, to be the primary determinant of end-user QoE. As intuitively expected, sustainable congestion impacts both QoS and QoE metrics, and is further amplified by large (bloated) buffers. When considering the opposite scenario (absence of congestion), we show that –even bloated– buffers impact QoS metrics. However, our results suggest that their impact on QoE metrics is in fact only marginal.

As one of the largest and most used *information systems* ever built, the Web runs on top of the largest computer *network*: the Internet. Fueled by its commercialization and the increasing number of services relying on it, the Web has gained substantial complexity, as manifested by the increase of data exchanges between automated hosts on the Internet. We show how these interactions occur at a large scale and also show how they support the three enablers for today's Web (*Web content delivery, monetization, and search*); thus becoming an essential component of today's Web. Finally, we also scrutinize how choices concerning Internet infrastructure and its protocols ultimately affect the Web. For example, we show how *ad-blockers* affect the *application layer* and block requests that support *content monetization* and by extension *content delivery and search*. We study the conditions under which protocols in the *transport layer* affect Web browsing performance; as well as protocols in the *network* layer affect the *reachability* of Web content.

7.2 Future work

The Web and the Internet are constantly evolving. This dissertation takes a step forward towards improving our understanding of this ecosystem and its increasing complexity. Given the insights previously discussed, we next discuss new research avenues.

Monetization. The rise of ad-blockers presents an unprecedented situation that affects they way content is published and consumed on the Web. Without clear regulation guidelines, advertisers and publishers may opt to deploy aggressive campaigns that increase revenue at the expense of user's privacy and QoE. This state of affairs calls for research in the QoE domain, traditionally focused on performance. In particular, we argue for field studies with human subjects with the objective to find a balance between effective advertisement strategies and low QoE impairments. Likewise, given the attention and concerns regarding the use of privacy-intrusive technology to improve revenue, future work calls for advancing techniques that enable privacy-respecting advertisement campaigns (e.g., [238]). In fact, a continuation of passive measurements on this area will allow us to inform regulatory bodies and the ongoing debate regarding online advertisements.

Search. Finding and indexing content hosted on the Web is a challenging task, as underlined by the small number of organizations that do so. Most of them monetize their services with advertisements (*front-office* Web traffic for *content search*). Thus, the arguments above also apply to them. That said, in this work we do not investigate how users interact with search services in detail. While we enumerate related work that reports on this topic (see §2.1.2), future work calls for a deeper investigation that complements this body of literature, e.g., how much content users consume via search results and how much by navigating Web links. This has implications for Web performance as being able to predict a user's click-stream enables performing DNS lookups earlier and pre-fetching content (thereby reducing page load times). Assessing potential performance gains is at the heart of this activity. The second direction of research concerns the ongoing debate about "the right to be forgotten" [104]. From a technical perspective, the debate affects Web indexes dynamics and Web crawlers. It calls for investigating how effective, when, and how some information is deleted from the Web and Web search services. Such an investigation will make it possible to provide informed guidelines to regulatory bodies. Finally, the advent of human-to-computer interfaces other than Web browsers (e.g., voice-commanded applications like Apple's Siri), opens the door for new research opportunities. Such interfaces introduce more complexity. For example, using these interfaces to implement e-commerce will require evolving the structure of Web requests, responses, and the Web data itself. We envision that it will further boost machine-to-server communication on the Internet. Perhaps, this ecosystem will lay the ground for the long-time envisioned but yet unachieved *semantic Web* [268].

Content delivery. Throughout this manuscript, we show that the Web is evolving towards a "split-architecture model", with a front- and a back-office. This shift originated at the need to improve performance and builds on the assumption that the network is the performance bottleneck. However, this is not always the case and techniques to improve latency for content delivery can also be applied to end hosts [100]. In fact, recent studies have already started to explore bottlenecks on the Web itself [229]. We suggest that there are even more research opportunities. For example, one opportunity involves revisiting the *split-browser* architecture [8, 148]. This architecture enables offloading parts of the browser's functionality to the edge of the network, thus avoiding the larger delays of access networks and also expensive computations at the end user's device. The ongoing deployment of cloud resources at the edge of access networks provides indeed a unique opportunity to explore this avenue with at least two objectives in mind. First, improve performance. Second, leverage virtualization to improve user's security and privacy. Indeed, network operators, i.e., ISPs, can supply CDNs or content providers with the infrastructure that is specifically tailored to handle back-office traffic. For ISPs, this opens up additional opportunities for monetization of services and also provide a better experience to their end users. But, this comes at a cost: ISPs must invest in and enable micro data centers or virtualized services [49, 149] in their networks to harness the opportunity. Thus, future work calls for investigation on the deployment of such infrastructure, traffic engineering policies to better accommodate the two different classes of traffic, and on SLAs targeted at organizations operating a back office.

Deployment of new protocols. Deploying new protocols or protocol variants may be very slow or even infeasible if it requires changes in the core infrastructure of the Internet or all end Internet end systems. For instance, consider the adoption of a new TCP variant or IPv6. However, restricting the change to one organization makes this otherwise infeasible effort much easier. Indeed, most Web service companies have the ability to roll out updates to their infrastructure and upgrade it on a regular basis. Thus, if end users connect only to the front-end servers and the front-end and back-end servers are managed by a single entity, it is feasible to deploy changes rapidly and optimize back-office Web communications; these changes are restricted to the servers involved in carrying the back-office traffic. Examples include but are not limited to use of persistent TCP connections between servers, IPv6, multi-path TCP [147], on-the-fly Web page assembly [206], object pre-fetching, delta encoding, and offloading of computations. Indeed, even the TCP version and parameters can be chosen according to the tasks [145], e.g., the initial window size can be increased. Moreover, it is possible to adopt new networking paradigms, such as Software Defined Networking [182], much more quickly to handle the back-office traffic. As most end-user communication involves some back-office communication, these improvements can directly affect the end-user experience. Lastly, more efficient use of the networking resources can also reduce the cost of content delivery.

Web performance and passive measurements. The immediate consequence of the Web becoming the "narrow waist" motivates further studies on Web QoE. Standards like WebRTC [89] enables browsers to engage in tasks other than fetching websites, e.g., voice calls, video chats, and even P2P file sharing. This technology presents a unique opportunity to study performance for these Web-based applications. We plan to further scrutinize Web performance issues using QoE metrics. At the same time, we advocate for a continuation of passive measurement studies that can shed light on performance bottlenecks caused by network conditions, e.g., by using traces collected at residential broadband networks we can gain insights on performance degradations resulting from poor WiFi performance at home networks, faulty or outdated CPEs, or misconfigured end-user devices.

Acknowledgments

Doing this Ph.D thesis has been an extremely educating and rewarding experience. Along with what the academic exercise itself carries, this experience has represented for me not only the opportunity to learn a lot from my peers in academia, but also about my own personal limits. There has been plenty of times where I have felt exhausted and frustrated, but also many satisfying and joyful moments. The following lines are for those to whom I am specially grateful for supporting me during this process.

Clara, thank you for your comprehension and patience: I could not have achieved it without your understanding. With you being by my side I have been able to complete my studies and enjoy a beautiful life in Berlin. Thank you Maria Jesús, Jordi, and Maria for your unconditional support and for helping me to renew my energies during my occasional visits to Barcelona.

I have been very fortunate to have three mentors who have taken great care of me. The first of them is my supervisor, Anja Feldmann. Anja gave me this fantastic opportunity and took care that my only concerns were related to research problems. She was always available for *data-crunching* and *paper-editing* sessions. These sessions were tough, but the best training you can have as a Ph.D. student. Nikolaos Chatzis advised me during the initial years of my studies. He introduced me to the exciting field of Internet measurement and advised me during several stages of my career. George has been like a coach at INET. He was always there, his door open, and ready to provide guidance and feedback on any topic.

I want to express my special gratitude to two of my collaborators: Oliver Hohlfeld and Philipp Richter. Their hard work and their support have made possible the work on which this thesis builds on. Oliver welcomed me in his own project without any reluctance when I first joined the group. I honestly did not expect to start my Ph.D. by hearing voice recordings impaired by different traffic distributions, but it turned our that I learned in the process a lot about how to build testbeds, set up traffic generators, and apply QoE models. Philipp has been also a great officemate. I really enjoyed our fruitful discussions about research as well as all other topics (which were a lot). He really helped to make the late-night shifts before the submissions really productive and fun, despite the stressful situations we were in. In the end, it was all about doing *something* with Internet data.

As a member of the INET group, I want to thank all my colleagues for the nice times we had outside the university, and also for their excellent team work during the realization of our duties as Ph.D. students.

List of Figures

List of Tables

Bibliography

[1] https://easylist-downloads.adblockplus.org/easylist.txt.

[2] https://easylist-downloads.adblockplus.org/easyprivacy.txt.

[3] Acceptable ads manifesto. https://acceptableads.org/en/.

[4] Adblock Plus. https://adblockplus.org/. Last accessed: 2 June 2016.

[5] Adblock Plus filters explained. hhttps://adblockplus.org/en/filter-cheatsheet.

[6] Allowing acceptable ads in Adblock Plus. https://adblockplus.org/en/acceptable-ads.

[7] Amazon, Google and Microsoft Escape Adblock Plus, for a price. http://www.engadget.com/2015/02/03/amazon-google-microsoft-adblock-plus/.

[8] Amazon Silk: Split browser architecture. http://docs.aws.amazon.com/silk/latest/developerguide/split-arch.html.

[9] AT&T high speed Internet business edition service level agreement. http://www.att.com/gen/general?pid=6622. Last accessed: 28 May 2016.

[10] Bufferbloat. http://www.bufferbloat.net/.

[11] Current implementation of AI_ADDRCONFIG considered harmful. https://fedoraproject.org/wiki/QA/Networking/NameResolution/ADDRCONFIG?rd=Networking/NameResolution/ADDRCONFIG. Fedora.

[12] Display LUMAscape. http://www.lumapartners.com/lumascapes/display-ad-tech-lumascape/.

[13] Doubleclick nearing privacy settlements. http://news.cnet.com/2100-1023-871654.html.

[14] EasyList statistics: August 2011. https://easylist.adblockplus.org/blog/2011/09/01/easylist-statistics:-august-2011.

[15] Electronic Frontier Foundation's privacy badger. https://www.eff.org/privacybadger.

[16] Emulex Endace DAG technology. http://www.emulex.com/. Last accessed: 7 September 2014.

[17] FTC regulation of behavioral advertising. https://en.wikipedia.org/wiki/FTC_regulation_of_behavioral_advertising. Last accessed: 02 June 2016.

[18] FTC staff report: Self-regulatory principles for online behavioral advertising. www.ftc.gov/opa/2009/02/behavad.shtm.

[19] Ghostery. https://www.ghostery.com/en/.

[20] Global latency and packet delivery SLA. http://www.verizonenterprise.com/terms/global_latency_sla.xml. Last accessed: 28 May 2016.

[21] Google Chrome store AdBlock Plus statistics. https://chrome.google.com/webstore/detail/adblock-plus/cfhdojbkjhnklbpkdaibdccddilifddb.

[22] Google developers: Real-time bidding protocol. https://developers.google.com/ad-exchange/rtb/start#basics. Google. Last accessed: 28 May 2016.

[23] Imagining the Internet. http://www.elon.edu/e-web/predictions/150/1960.xhtml. Elon University School of Communications. Last accessed: 28 May 2016.

[24] Internet Advertising Bureau (IAB). 2013 Internet Advertising Revenue Report. http://www.iab.net/AdRevenueReport.

[25] libadblockplus: A C++ Library offering the core functionality of Adblock Plus. https://github.com/adblockplus/libadblockplus.

[26] Most popular extensions - Add-ons for Firefox. ttps://addons.mozilla.org/en-us/firefox/extensions/?sort=users.

[27] Mozilla AdBlock Plus statistics. https://addons.mozilla.org/en-US/firefox/addon/adblock-plus/statistics/usage/?last=30.

[28] Netflix Open Connect. https://signup.netflix.com/openconnect.

[29] NTT SLA: Global IP network. http://www.us.ntt.net/support/sla/network.cfm. NTT. Last accessed: 28 May 2016.

[30] PeeringDB. https://www.peeringdb.com. Last accessed: 28 May 2016.

[31] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. https://www.planet-lab.org/.

[32] Project BISmark. http://projectbismark.net/. Last accessed: 28 May 2016.

[33] Publishers watch closely as adoption of ad blocking tech grows. http://adage.com/article/digital/adoption-ad-blocking-tech-grows/297101/.

[34] RIPE Atlas. https://atlas.ripe.net/. Last accessed: 28 May 2016.

[35] SamKnows. https://www.samknows.com/. Last accessed: 28 May 2016.

[36] Selenium: Web Browser Automation. seleniumhq.org.

[37] SpeedTest. https://www.speedtest.net/. Last accessed: 28 May 2016.

[38] The World in 2014: ICT facts and figures. International Telecommunication Union. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf. Last accessed: 10 Dec. 2015.

[39] To crawl or not to crawl, that is BingBot's question. https://blogs.bing.com/webmaster/2012/05/03/to-crawl-or-not-to-crawl-that-is-bingbots-question/. Bing blogs.

[40] Web banner. https://en.wikipedia.org/wiki/Web_banner#History. Last accessed: 02 June 2016.

[41] Writing Adblock Plus Filters. https://adblockplus.org/en/filters.

[42] ITU-T recommendation P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, 2001.

[43] ITU-T recommendation G.107: The E-model, a computational model for use in transmission planning, 2003.

[44] ITU-T Rec. P.862 annex a: Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 a. P.862.2. http://www.itu.int/rec/T-REC-P.862-200511-I!Amd2/en, 2005.

[45] ITU-T Recommendation G.1030: estimating end-to-end performance in IP networks for data applications, 2005.

[46] ITU-T E.800: Definitions of terms related to quality of service, 2008. International Telecommunication Union (ITU).

[47] Global Internet user survey. http://www.internetsociety.org/internet/global-internet-user-survey-2012, 2012. Internet Society (ISOC). Last accessed: 1 February 2016.

[48] Qualinet White paper on Definitions of Quality of Experience. European Network on Quality of Experience in Multimedia Systems and Services, P. Le Callet, S. Möller and A. Perkis, eds., Version 1.1, June 2012.

[49] Network functions virtualisation (NFV), 2013. ETSI GS NFV 001 V1.1.1.

[50] Cisco visual networking index: Forecast and methodology, 2014-2019. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf, 2015. Cisco.

[51] comScore releases September 2015 U.S. desktop search engine rankings. http://www.comscore.com/Insights/Market-Rankings/comScore-Releases-September-2015-US-Desktop-Search-Engine-Rankings, 2015. comScore, Inc.

[52] Global Internet Phenomena - Africa, Middle East & North America, 2015. Sandvine Intelling Broadband Networks.

[53] Google Will Take 55% of Search Ad Dollars Globally in 2015. http://www.emarketer.com/Article/Google-Will-Take-55-of-Search-Ad-Dollars-Globally-2015/1012294?ecid=PR1016, 2015. eMarketer.

[54] Amsterdam Internet Exchange IPv6 Traffic. https://ams-ix.net/technical/statistics/sflow-stats/ipv6-traffic, 2016.

[55] ARIN IPv6 Wiki: Broadband CPE. https://getipv6.info/display/IPv6/Broadband+CPE, 2016.

[56] IPv6 - Google. http://www.google.com/intl/en/ipv6/statistics.html, 2016.

[57] Learn about robots.txt files. https://support.google.com/webmasters/answer/6062608, 2016. Google.

[58] Market share for mobile, browsers, operating systems and search engines. http://www.netmarketshare.com/, 2016. NetMarketShare.

[59] World IPv6 Launch. http://www.worldipv6launch.org/, 2016.

[60] ABEN, E. Hampering eyeballs - Observations on two "happy eyeballs" implementations. https://labs.ripe.net/Members/emileaben/hampered-eyeballs.

[61] ABEN, E., TRENAMAN, N., KIESSLING, A., AND WILHELM, R. Lost starts - Why operators switch off IPv6, 2016. NANOG 66.

[62] ABOBA, B., ZORN, G., AND MITTON, D. RADIUS and IPv6. RFC 3162 (Proposed Standard), Aug. 2001.

[63] ADHIKARI, V. K., JAIN, S., CHEN, Y., AND ZHANG, Z. L. Vivisecting YouTube: An active measurement study. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2012).

[64] AGER, B., CHATZIS, N., FELDMANN, A., SARRAR, N., UHLIG, S., AND WILLINGER, W. Anatomy of a large European IXP. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2012).

[65] AGER, B., MUHLBAUER, W., SMARAGDAKIS, G., AND UHLIG, S. Web content cartography. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[66] AGER, B., SCHNEIDER, F., KIM, J., AND FELDMANN, A. Revisiting cacheability in times of user generated content. In *Proceedings of the IEEE Global Internet Symposium (GI)* (2010).

[67] ALBASIR, A., NAIK, K., PLOURDE, B., AND GOEL, N. Experimental study of energy and bandwidth costs of Web advertisements on smartphones. In *Proceedings of the EAI International Conference on Mobile Computing, Applications and Services (MobiCASE)* (2014).

[68] ALCOCK, S., LORIER, P., AND NELSON, R. Libtrace: A packet capture and analysis library. *ACM SIGCOMM Computer Communication Review (CCR) 42*, 2 (2012).

[69] ALLMAN, M. Comments on bufferbloat. *ACM SIGCOMM Computer Communication Review (CCR) 43*, 1 (2012).

[70] ALZOUBI, H. A., LEE, S., RABINOVICH, M., SPATSCHECK, O., AND VAN DER MERWE, J. A practical architecture for an anycast CDN. *ACM Transactions on the Web 5*, 4 (2011).

[71] ANDREWS, M. Negative caching of DNS Queries (DNS NCACHE). RFC 2308 (Proposed Standard), Mar. 1998. Updated by RFCs 4035, 4033, 4034, 6604.

[72] ANGEL, S., AND WALFISH, M. Verifiable auctions for online ad exchanges. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2013).

[73] APPENZELLER, G., KESLASSY, I., AND MCKEOWN, N. Sizing router buffers. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2004).

[74] B. Chandrasekaran and G. Smaragdakis and A. Berger and M. Luckie and K. Ng. A server-to-server view of the Internet. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2015).

[75] Bajpai, V., Eravuchira, S. J., and Schönwälder, J. Lessons learned from using the RIPE Atlas platform for measurement research. *ACM SIGCOMM Computer Communication Review (CCR) 45*, 3 (July 2015).

[76] Bajpai, V., and Schönwälder, J. IPv4 versus IPv6 - Who connects faster? In *Proceedings of the IFIP Networking conference (NETWORKING)* (2015).

[77] Balebako, R., Leon, P., Shay, R., Ur, B., Wang, Y., and Cranor, L. Measuring the effectiveness of privacy tools for limiting behavioral advertising. In *Proceedings of the IEEE Workshop on Web 2.0 Security and Privacy (W2SP)* (2012).

[78] Barakat, N., and Darcie, T. E. Delay characterization of cable access networks. *IEEE Communications Letters 11*, 4 (2007).

[79] Barford, P., Canadi, I., Krushevskaja, D., Ma, Q., and Muthukrishnan, S. Adscape: Harvesting and Analyzing Online Display Ads. In *Proceedings of the International World Wide Web conference (WWW)* (2014).

[80] Barish, G., and Obraczke, K. World Wide Web caching: Trends and techniques. *IEEE Communications magazine 38*, 5 (2000).

[81] Barroso, L. A., Dean, J., and Holzle, U. Web search for a planet: The Google clustering architecture. *IEEE Micro 23*, 2 (2003).

[82] Bauer, S., Clark, D. D., and Lehr, W. Understanding broadband speed measurements. In *Proceedings of the Research Conference on Communications, Information and Internet Policy (TPRC)* (2010).

[83] Beheshti, N., Ganjali, Y., Ghobadi, M., McKeown, N., and Salmon, G. Experimental study of router buffer sizing. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2008).

[84] Belshe, M. More bandwidth doesn't matter (much). https://goo.gl/f6ngQ.

[85] Belshe, M., Peon, R., and Thomson, M. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540 (Proposed Standard), May 2015.

[86] Ben Houidi, Z., Scavo, G., Ghamri-Doudane, S., Finamore, A., Traverso, S., and Mellia, M. Gold mining in a river of Internet content traffic. In *Proceedings of the International conference on Traffic Monitoring and Analysis (TMA)* (2014).

[87] Benson, T., Akella, A., and Maltz, D. A. Network traffic characteristics of data centers in the wild. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[88] Benson, T., Anand, A., Akella, A., and Zhang, M. MicroTE: Fine grained traffic engineering for data centers. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2011).

[89] Bergkvist, A., Burnett, D. C., Jennings, C., Narayanan, A., and Aboba, B. WebRTC 1.0: Real-time communication between browsers. W3C Editor's Draft, May 2016. W3C.

[90] Bermudez, I. N., Mellia, M., Munafò, M., Keralapura, R., and Nucci, A. DNS to the rescue: Discerning content and services in a tangled Web. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[91] Bernaille, L., and Teixeira, R. Early recognition of encrypted applications. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2007).

[92] Berners-Lee, T. The World Wide Web: Past, present and future. https://www.w3.org/People/Berners-Lee/1996/ppf.html.

[93] Berners-Lee, T., and Cailliau, R. WorldWideWeb: Proposal for a HyperText project. https://www.w3.org/Proposal.html, 1990. W3C. Last accessed: 28 May 2016.

[94] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A. The World-Wide Web. *Communications of the ACM 37*, 8 (1994).

[95] Berners-Lee, T., Fielding, R., and Frystyk, H. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945 (Informational), May 1996.

[96] Beverly, R., Luckie, M., Mosley, L., and Claffy, K. Measuring and characterizing IPv6 router availability. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2015).

[97] Bixby, J. 4 awesome slides showing how page speed correlates to business metrics at Walmart.com. http://www.webperformancetoday.com/2012/02/28/4-awesome-slides-showing-how-page-speed-correlates-to-business-metrics\/-at-walmart-com/. Web Performance Today.

[98] Book, T., and Wallach, D. S. An empirical study of mobile ad targeting. *arXiv 1502.06577* (2015).

[99] Brin, S., and Page, L. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the International World Wide Web conference (WWW)* (1998).

[100] Briscoe, B., Brunstrom, A., Petlund, A., Hayes, D., Ros, D., Tsang, I. J., Gjessing, S., Fairhurst, G., Griwodz, C., and Welzl, M. Reducing Internet latency: A survey of techniques and their merits. *IEEE Communications Surveys Tutorials* (2014).

[101] Bush, V., and Wang, J. As we may think. *Atlantic Monthly 176* (1945).

[102] Butkiewicz, M., Madhyastha, H. V., and Sekar, V. Understanding Website complexity: Measurements, metrics, and implications. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[103] Butkiewicz, M., Madhyastha, H. V., and Sekar, V. Characterizing Web page complexity and its impact. *IEEE/ACM Transactions on Networking 22*, 3 (2014).

[104] Bygrave, L. A. A right to be forgotten? *Communications of the ACM 58*, 1 (2014).

[105] Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., and Govindan, R. Mapping the expansion of Google's serving infrastructure. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[106] Callahan, T., Allman, M., and Rabinovich, M. On modern DNS behavior and properties. *ACM SIGCOMM Computer Communication Review (CCR) 43*, 3 (2013).

[107] CANADI, I., BARFORD, P., AND SOMMERS, J. Revisiting broadband performance. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[108] CANTÓ PALANCAR, R., LÓPEZ DA SILVA, R. A., FOLGUEIRA CHAVARRÍA, J. L., LÓPEZ, D. R., ELIZONDO ARMENGOL, A. J., AND GAMERO TINOCO, R. Virtualization of residential customer premise equipment. Lessons learned in Brazil vCPE trial. *Information Technology 57*, 5 (2015).

[109] CASTILLO, C. Effective Web crawling. *ACM SIGIR Forum 39*, 1 (2005).

[110] CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W. C., WALLACH, D. A., BURROWS, M., CHANDRA, T., FIKES, A., AND GRUBER, R. E. Bigtable: A distributed storage system for structured data. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)* (2006).

[111] CHATZIS, N., SMARAGDAKIS, G., FELDMANN, A., AND WILLINGER, W. There is more to IXPs than meets the eye. *ACM SIGCOMM Computer Communication Review (CCR) 43*, 5 (2013).

[112] CHEN, X., JINDAL, A., AND HU, Y. C. How much energy can we save from prefetching ads?: Energy drain analysis of top 100 apps. In *Proceedings of the ACM Workshop on Power Aware Computing and Systems (HotPower)* (2013).

[113] CHEN, Y., JAIN, S., ADHIKARI, V. K., AND ZHANG, Z. L. Characterizing roles of front-end servers in end-to-end performance of dynamic content distribution. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[114] CHEN, Y., MAHAJAN, R., SRIDHARAN, B., AND ZHANG, Z. L. A Provider-side view of Web search response time. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2013).

[115] CHIRICHELLA, C., AND ROSSI, D. To the moon and back: Are Internet bufferbloat delays really that large? In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2013).

[116] CHIU, Y.-C., SCHLINKER, B., RADHAKRISHNAN, A. B., KATZ-BASSETT, E., AND GOVINDAN, R. Are we one hop away from a better internet? In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2015).

[117] CHO, J., AND GARCIA-MOLINA, H. The evolution of the Web and implications for an incremental crawler. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)* (2000).

[118] CHO, K., LUCKIE, M., AND HUFFAKER, B. Identifying IPv6 network problems in the dual-stack world. In *Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality (NetT)* (2004).

[119] CLAFFY, K. Tracking IPv6 evolution: Data we have and data we need. *ACM SIGCOMM Computer Communication Review (CCR) 41*, 3 (2011).

[120] CLARK, D., BAUER, S., CLAFFY, K., DHAMDHERE, A., HUFFAKER, B., LEHR, W., AND LUCKIE, M. Measurement and analysis of Internet interconnection and congestion. In *Proceedings of the Research Conference on Communications, Information and Internet Policy (TPRC)* (2014).

[121] Colitti, L., Gunderson, S. H., Kline, E., and Refice, T. Evaluating IPv6 adoption in the Internet. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2010).

[122] Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., and Bailey, M. Measuring IPv6 adoption. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2014).

[123] Damas, J., Graff, M., and Vixie, P. Extension Mmchanisms for DNS (EDNS(0)). RFC 6891 (Internet Standard), Apr. 2013.

[124] D'Ambrosio, S., De Pasquale, S., Iannone, G., Malandrino, D., Negro, A., Patimo, G., Petta, A., Scarano, V., Serra, L., and Spinelli, R. Mobile phone batteries draining: Is green Web browsing the solution? In *Proceedings of the International Green Computing Conference (IGCC)* (2014).

[125] Datta, A., Tschantz, M. C., and Datta, A. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings on Privacy Enhancing Technologies (PETS)* (2014).

[126] Davison, B. D. A web caching primer. *IEEE Internet Computing 5*, 4 (July 2001).

[127] Dean, J., and Ghemawat, S. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008).

[128] Dec, W., Sarikaya, B., Zorn, G., Miles, D., and Lourdelet, B. RADIUS attributes for IPv6 access networks. RFC 6911 (Proposed Standard), Apr. 2013.

[129] Dhamdhere, A., Luckie, M., Huffaker, B., Claffy, K., Elmokashfi, A., and Aben, E. Measuring the deployment of IPv6: Topology, routing and performance. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[130] Dischinger, M., Haeberlen, A., Gummadi, K. P., and Saroiu, S. Characterizing residential broadband networks. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2007).

[131] Dobrian, F., Awan, A., Joseph, D., Ganjam, A., Zhan, J., Sekar, V., Stoica, I., and Zhang, H. Understanding the impact of video quality on user engagement. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2011).

[132] Douglis, F., Feldmann, A., Krishnamurthy, B., and Mogul, J. Rate of Change and Other Metrics: A Live Study of the World Wide Web. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)* (1997).

[133] Drago, I., an M. Munafo, M. M., Sperotto, A., Sadre, R., and Pras, A. Inside Dropbox: Understanding personal cloud storage services. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[134] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-wide scanning and its security applications. In *Proceedings of the USENIX Security Symposium* (2013).

[135] Egger, S., Hossfeld, T., Schatz, R., and Fiedler, M. Tutorial: Waiting Times in Quality of Experience for Web based Services. In *Proceedings of the IEEE International Workshop on Quality of Multimedia Experience (QoMEX)* (2012).

[136] EGGER, S., REICHL, P., HOSFELD, T., AND SCHATZ, R. Time is bandwidth? Narrowing the gap between subjective time perception and Quality of Experience. In *Proceedings of the IEEE International Conference on Communications (ICC)* (2012).

[137] EGGER, S., SCHATZ, R., SCHOENENBERG, K., RAAKE, A., AND KUBIN, G. Same but different? - Using speech signal features for comparing conversational VoIP quality studies. In *Proceedings of the IEEE International Conference on Communications (ICC)* (2012).

[138] ENACHESCU, M., GANJALI, Y., GOEL, A., MCKEOWN, N., AND ROUGHGARDEN, T. Routers with very small buffers. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2006).

[139] ERMAN, J., GERBER, A., HAJIAGHAYI, M., PEI, D., AND SPATSCHECK, O. Network-aware forward caching. In *Proceedings of the International World Wide Web conference (WWW)* (2009).

[140] FARAHAT, A., AND BAILEY, M. C. How effective is targeted advertising? In *Proceedings of the International World Wide Web conference (WWW)* (2012).

[141] FEAMSTER, N. Revealing utilization at Internet interconnection points. *arXiv:1603.03656* (2016).

[142] FELDMANN, A., KAMMENHUBER, N., MAENNEL, O., MAGGS, B., PRISCO, R. D., AND SUNDARAM, R. A methodology for estimating interdomain Web traffic demand. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2004).

[143] FETTERLY, D., MANASSE, M., NAJORK, M., AND WIENER, J. A large-scale study of the evolution of Web pages. In *Proceedings of the International World Wide Web conference (WWW)* (2003).

[144] FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Obsoleted by RFCs 7230, 7231, 7232, 7233, 7234, 7235, updated by RFCs 2817, 5785, 6266, 6585.

[145] FLACH, T., DUKKIPATI, N., TERZIS, A., RAGHAVAN, B., CARDWELL, N., CHENG, Y., JAIN, A., HAO, S., KATZ-BASSETT, E., AND GOVINDAN, R. Reducing Web latency: The virtue of gentle aggression. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2013).

[146] FLAVEL, A., MANI, P., MALTZ, D. A., HOLT, N., LIU, J., CHEN, Y., AND SURMACHEV, O. FastRoute: A scalable load-aware anycast routing architecture for modern CDNs. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2015).

[147] FORD, A., RAICIU, C., HANDLEY, M., BARRE, S., AND IYENGAR, J. Architectural guidelines for multipath TCP development. RFC 6182 (Informational), Mar. 2011.

[148] FOX, A., GOLDBERG, I., GRIBBLE, S. D., LEE, D. C., POLITO, A., AND BREWER, E. A. Experience with top gun wingman: A proxy-based graphical Web browser for the 3Com PalmPilot. In *Proceedings of IFIP Middleware* (1998).

[149] FRANK, B., POESE, I., LIN, Y., SMARAGDAKIS, G., FELDMANN, A., MAGGS, B., RAKE, J., UHLIG, S., AND WEBER, R. Pushing CDN-ISP collaboration to the limit. *ACM SIGCOMM Computer Communication Review (CCR) 43*, 3 (2013).

[150] G, J., Jiang, J., Duan, H., Li, K., Wan, T., and Wu, J. When HTTPS meets CDN: A case of authentication in delegated service. In *Proceedings of the IEEE Symposium on Security and Privacy (SSP)* (2014).

[151] Gao, H., Yegneswaran, V., Chen, Y., Porras, P., Ghosh, S., Jiang, J., and Duan, H. An empirical reexamination of global DNS behavior. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2013).

[152] Gerber, A., and Doverspike, R. Traffic types and growth in backbone networks. In *Proceedings of the Optical Fiber Communication Conference (OFC)* (2011).

[153] Gettys, J. IW10 considered harmful. IETF Internet-Draft, 2011.

[154] Gettys, J., and Nichols, K. Bufferbloat: Dark buffers in the Internet. *ACM Queue 9*, 11 (2011).

[155] Gill, P., Arlitt, M., Li, Z., and Mahanti, A. The flattening Internet topology: natural evolution, unsightly barnacles or contrived collapse? In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2008).

[156] Gill, P., Erramilli, V., Chaintreau, A., Krishnamurthy, B., Papagiannaki, K., and Rodriguez, P. Follow the money: Understanding economics of online aggregation and advertising. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[157] Gill, P., Erramilli, V., Chaintreau, A., Krishnamurthy, B., Papagiannaki, K., and Rodríguez, P. Follow the Money: Understanding Economics of Online Aggregation and Advertising. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[158] Giotsas, V., Luckie, M., Huffaker, B., and Claffy, K. IPv6 AS relationships, clique, and congruence. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2015).

[159] Goga, O., and Teixeira, R. Speed measurements of residential Internet access. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2012).

[160] Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. VL2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2009).

[161] Gu, Y., Towsley, D. F., Hollot, C. V., and Zhang, H. Congestion control for small buffer high speed networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2007).

[162] Gugelmann, D., Happe, M., Ager, B., and Lenders, V. An automated approach for complementing ad blockers blacklists. In *Proceedings on Privacy Enhancing Technologies (PETS)* (2015).

[163] Guha, S., Cheng, B., and Francis, P. Challenges in measuring online advertising systems. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[164] Guha, S., Cheng, B., and Francis, P. Privad: Practical privacy in online advertising. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2011).

[165] GUMMADI, K. P., SAROIU, S., AND GRIBBLE, S. D. King: Estimating latency between arbitrary Internet end hosts. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop* (2002).

[166] GUSE, D., EGGER, S., RAAKE, A., AND MÖLLER, S. Web-QoE under real-world distractions: Two test cases. In *Proceedings of the IEEE International Workshop on Quality of Multimedia Experience (QoMEX)* (2014).

[167] GYSI, M. Status of Swisscom's IPv6 activities, outlook and opportunities. In *Proceedings of the Swiss IPv6 Council IPv6 Business Conference* (2016).

[168] HADDADI, H., HUI, P., HENDERSON, T., AND BROWN, I. Targeted advertising on the handset: Privacy and security challenges. In *Pervasive Advertising*. 2011.

[169] HAIQING JIANG, YAOGONG WANG, K. L., AND RHEE, I. Tackling bufferbloat in 3G/4G networks. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[170] HALEPOVIC, E., PANG, J., AND SPATSCHECK, O. Can you GET me now?: Estimating the time-to-first-byte of HTTP transactions with passive measurements. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[171] HARDT, M., AND NATH, S. Privacy-aware Personalization for Mobile Advertising. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (2012).

[172] HEUSSE, M., MERRITT, S. A., BROWN, T. X., AND DUDA, A. Two-way TCP connections: old problem, new insight. *ACM SIGCOMM Computer Communication Review (CCR) 41*, 2 (2011).

[173] HOGEWONING, M. IPv6 CPE Overview. http://ripe60.ripe.net/presentations/Hogewoning-IPv6_CPE_overview.pdf, 2010. RIPE 60.

[174] HOHLFELD, O., BALARAJAH, B., BENNER, S., RAAKE, A., AND CIUCU, F. On revealing the ARQ mechanism of MSTV. In *Proceedings of the IEEE International Conference on Communications (ICC)* (2011).

[175] HOHLFELD, O., PUJOL, E., CIUCU, F., FELDMANN, A., AND BARFORD, P. A QoE perspective on sizing network buffers. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2014).

[176] HOLTERBACH, T., PELSSER, C., BUSH, R., AND VANBEVER, L. Quantifying interference between measurements on the RIPE Atlas platform. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2015).

[177] HUANG, C., WANG, A., LI, J., AND ROSS, K. Measuring and evaluating large-scale CDNs. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2008).

[178] HUSTON, G. Bemused eyeballs. https://labs.apnic.net/?p=188, 2012.

[179] HUSTON, G. Revisiting Apple and IPv6. https://blog.apnic.net/2015/07/15/revisiting-apple-and-ipv6/, 2015.

[180] IHM, S., AND PAI, V. S. Towards understanding modern Web traffic. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[181] JACOBSON, V. Modified TCP congestion control algorithm. end2end-interest mailing list, Apr. 1990.

[182] JAIN, S., KUMAR, A., MANDAL, S., ONG, J., POUTIEVSKI, L., SINGH, A., VENKATA, S., WANDERER, J., ZHOU, J., ZHU, M., ZOLLA, J., HOLZLE, U., STUART, S., AND VAHDAT, A. B4: Experience with a globally-deployed software defined WAN. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2013).

[183] JIANG, H., AND DOVROLIS, C. Passive estimation of TCP round-trip times. *ACM SIGCOMM Computer Communication Review (CCR) 32*, 3 (2002).

[184] KAMMENHUBER, N., LUXENBURGER, J., FELDMANN, A., AND WEIKUM, G. Web search clickstreams. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2006).

[185] KARIR, M., HUSTON, G., MICHAELSON, G., AND BAILEY, M. Understanding IPv6 populations in the wild. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2013).

[186] KARPILOVSKY, E., GERBER, A., PEI, D., REXFORD, J., AND SHAIKH, A. Quantifying the extent of IPv6 deployment. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2009).

[187] KIM, J., SARRAR, N., AND FELDMANN, A. Watching the IPv6 takeoff from an IXP's viewpoint. *arXiv:1402.3982* (2014).

[188] KITAWAKI, N., AND ITOH, K. Pure delay effects on speech quality in telecommunications. *IEEE Journal on Selected Areas in Communications 9*, 4 (1991).

[189] KLEINBERG, J. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM/SIAM Symposium on Discrete Algorithms (SODA)* (1998).

[190] KOEHL, A., AND WANG, H. Surviving a search engine overload. In *Proceedings of the International World Wide Web conference (WWW)* (2012).

[191] KOHAVI, R., HENNE, R. M., AND SOMMERFIELD, D. Practical guide to controlled experiments on the Web: Listen to your customers not to the HiPPO. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)* (2007).

[192] KONTAXIS, G., AND CHEW, M. Tracking protection in Firefox for privacy and performance. *arXiv 1506.04104* (2015).

[193] KORHONEN, J., AND YOU, J. Peak signal-to-noise radio revised: Is simple beautiful? In *Proceedings of the IEEE International Workshop on Quality of Multimedia Experience (QoMEX)* (2012).

[194] KOROLOVA, A. Privacy violations using microtargeted ads: A case study. In *Proceedings of the IEEE International Workshop on Privacy Aspects of Data Mining (PADM)* (2010).

[195] KREIBICH, C., WEAVER, N., MAIER, G., NECHAEV, B., AND PAXSON, V. Experiences from Netalyzr with engaging users in end-system measurement. In *Proceedings of the ACM SIGCOMM Workshop on Measurements up the Stack (W-MUST)* (2011).

[196] KREIBICH, C., WEAVER, N., NECHAEV, B., AND PAXSON, V. Netalyzr: Illuminating the edge network. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[197] KRISHNAMURTHY, B., AND REXFORD, J. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Addison-Wesley, 2001.

[198] KRISHNAMURTHY, B., AND WILLS, C. Privacy diffusion on the Web: A longitudinal perspective. In *Proceedings of the International World Wide Web conference (WWW)* (2009).

[199] KRISHNAMURTHY, B., AND WILLS, C. E. Cat and mouse: Content delivery tradeoffs in Web access. In *Proceedings of the International World Wide Web conference (WWW)* (2006).

[200] KRISHNAN, R., MADHYASTHA, H., SRINIVASAN, S., JAIN, S., KRISHNAMURTHY, A., ANDERSON, T., AND GAO, J. Moving beyond end-to-end path information to optimize CDN performance. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[201] KRISHNAN, S. S., AND SITARAMAN, R. K. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[202] KUROSE, J. F., AND ROSS, K. *Computer Networking: A Top-Down Approach Featuring the Internet*, 2nd ed. Addison-Wesley, 2002.

[203] LABOVITZ, C., LEKEL-JOHNSON, S., MCPHERSON, D., OBERHEIDE, J., AND JAHANIAN, F. Internet inter-domain traffic. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2010).

[204] LAGERHOLM, S., AND ROSELLI, J. Negative caching of DNS records, 2015. Microsoft.

[205] LAKSHMIKANTHA, A., BECK, C., AND SRIKANT, R. Impact of file arrivals and departures on buffer sizing in core routers. *IEEE/ACM Transactions on Networking 19*, 2 (Apr. 2011).

[206] LEIGHTON, T. Improving performance on the Internet. *Communications of the ACM 52*, 2 (2009).

[207] LEINER, B. M., CERF, V. G., CLARK, D. D., KAHN, R. E., KLEINROCK, L., LYNCH, D. C., POSTEL, J., ROBERTS, L. G., AND WOLFF, S. Brief history of the Internet. http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet. Internet Society (ISOC).

[208] LI, Z., ZHANG, K., XIE, Y., YU, F., AND WANG, X. Knowing your enemy: Understanding and detecting malicious Web advertising. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (2012).

[209] LIDEN, G. Marissa Mayer at Web 2.0. http://glinden.blogspot.de/2006/11/marissa-mayer-at-web-20.html, 2006.

[210] LINDEN, G. Make data useful. http://www.gduchamp.com/media/StanfordDataMining.2006-11-28.pdf, 2006.

[211] LIVADARIU, I., ELMOKASHFI, A., AND DHAMDHERE, A. Characterizing IPv6 control and data plane stability. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2016).

[212] LIVADARIU, I., ELMOKASHFI, A., DHAMDHERE, A., AND CLAFFY, K. A first look at IPv4 transfer markets. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2013).

[213] LODHI, A., LARSON, N., DHAMDHERE, A., DOVROLIS, C., AND CLAFFY, K. Using peeringDB to understand the peering ecosystem. *ACM SIGCOMM Computer Communication Review (CCR) 44*, 2 (2014).

[214] LUCKIE, M., BEVERLY, R., BRINKMEYER, W., AND CLAFFY, K. Speedtrap: Internet-scale IPv6 alias resolution. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[215] MADHAVAN, J., KO, D., KOT, L., GANAPATHY, V., RASMUSSEN, A., AND HALEVY, A. Google's deep web crawl. *VLDB Endowment 1*, 2 (Aug. 2008).

[216] MAIER, G., FELDMANN, A., PAXSON, V., AND ALLMAN, M. On dominant characteristics of residential broadband Internet traffic. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[217] MAIER, G., SCHNEIDER, F., AND FELDMANN, A. NAT usage in residential broadband networks. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2011).

[218] MARTIN, J., WESTALL, J., SHAW, T., WHITE, G., WOUNDY, R., FINKELSTEIN, J., AND HART, G. Cable modem buffer management in docsis networks. In *Proceedings of the IEEE Sarnoff Symposium* (2010).

[219] MCCONACHIE, A. How to make your Website available over IPv6. http://www.internetsociety.org/deploy360/blog/2014/06/how-to-make-your-website-available-over-ipv6/, 2014.

[220] METWALLEY, H., TRAVERSO, S., AND MELLIA, M. Using passive measurements to demystify online trackers. *IEEE Computer 49*, 3 (2016).

[221] METWALLEY, H., TRAVERSO, S., MELLIA, M., MISKOVIC, S., AND BALDI, M. The online tracking horde: A view from passive measurements. In *Proceedings of the International conference on Traffic Monitoring and Analysis (TMA)* (2015).

[222] MISRA, V. Routing money, not packets. *Communications of the ACM 58*, 6 (2015), 24–27.

[223] MÖLLER, S. *Quality of telephone-based spoken dialogue systems*, 1st ed. Springer, 2010.

[224] MÖLLER, S., CHAN, W.-Y., CÔTÉ, N., FALK, T. H., RAAKE, A., AND WÄLTERMANN, M. Speech quality estimation: Models and trends. *IEEE Signal Processing Magazine 28*, 6 (2011).

[225] MÖLLER, S., AND RAAKE, A., Eds. *Quality of Experience: Advanced Concepts, Applications and Methods*. Springer, 2014.

[226] MORI, T., INOUE, T., SHIMODA, A., SATO, K., ISHIBASHI, K., AND GOTO, S. SFMap: Inferring Services over Encrypted Web Flows Using Dynamical Domain Name Graphs. In *Proceedings of the International conference on Traffic Monitoring and Analysis (TMA)*. 2015.

[227] MORISHITA, Y., AND JINMEI, T. Common misbehavior against DNS queries for IPv6 addresses. RFC 4074 (Informational), May 2005.

[228] MUGHEES, M. H., QIAN, Z., SHAFIQ, Z., DASH, K., AND HUI, P. A first look at ad-block detection: A new arms race on the Web. *arXiv:1605.05841* (2016).

[229] NARAYANAN, S. P., NAM, Y. S., SIVAKUMAR, A., CHANDRASEKARAN, B., MAGGS, B., AND RAO, S. Reducing latency through page-aware management of Web objects by content delivery networks. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)* (2016).

[230] NATH, S. MAdScope: Characterizing mobile in-app targeted ads. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)* (2015).

[231] NAYLOR, D., FINAMORE, A., LEONTIADIS, I., GRUNENBERGER, Y., MELLIA, M., MUNAFÒ, M., PAPAGIANNAKI, K., AND STEENKISTE, P. The cost of the S in HTTPS. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2014).

[232] NICHOLS, K., AND JACOBSON, V. Controlling queue delay. *ACM Queue 10*, 5 (May 2012).

[233] NIKKHAH, M., AND GUÉRIN, R. Migrating the Internet to IPv6: An exploration of the when and why. *IEEE/ACM Transactions on Networking* (2015).

[234] NIKKHAH, M., GUÉRIN, R., LEE, Y., AND WOUNDY, R. Assessing IPv6 through Web access a measurement study and its findings. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2011).

[235] NYGREN, E., SITARAMAN, R. K., AND SUN, J. The Akamai network: A platform for high-performance Internet applications. *ACM SIGOPS Operating Systems Review (OSR)* (2010).

[236] OSTERMANN, S. Tcptrace, 2005. http://www.tcptrace.org/.

[237] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. The PageRank citation ranking: Bringing order to the Web. Stanford InfoLab, 1999.

[238] PAPAODYSSEFS, F., IORDANOU, C., BLACKBURN, J., LAOUTARIS, N., AND PAPAGIANNAKI, K. Web identity translator: Behavioral advertising and identity privacy with WIT. In *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)* (2015).

[239] PAXSON, V. Bro: A system for detecting network intruders in real-time. *Computer Networks 31*, 23 (1999).

[240] PLONKA, D., AND BARFORD, P. Context-aware clustering of DNS query traffic. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2008).

[241] PLONKA, D., AND BARFORD, P. Assessing performance of Internet services on IPv6. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)* (2013).

[242] PODLIPNIG, S., AND BÖSZÖRMENYI, L. A Survey of Web cache replacement strategies. *ACM Computing Surveys 35*, 4 (2003).

[243] POESE, I., FRANK, B., AGER, B., SMARAGDAKIS, G., AND FELDMANN, A. Improving content delivery using provider-aided distance information. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[244] POESE, I., FRANK, B., SMARAGDAKIS, G., UHLIG, S., FELDMANN, A., AND MAGGS, B. Enabling content-aware traffic engineering. *ACM SIGCOMM Computer Communication Review (CCR) 42*, 5 (2012).

[245] POPA, L., GHODSI, A., AND STOICA, I. HTTP as the narrow waist of the future Internet. In *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)* (2010).

[246] PRASAD, R. S., DOVROLIS, C., AND THOTTAN, M. Router buffer sizing for TCP traffic and the role of the output/input capacity ratio. *IEEE/ACM Transactions on Networking 17*, 5 (Oct. 2009).

[247] PUJOL, E., HOHLFELD, O., AND FELDMANN, A. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2015).

[248] PUJOL, E., RICHTER, P., CHANDRASEKARAN, B., SMARAGDAKIS, G., FELDMANN, A., MAGGS, B., AND NG, K. C. Back-office Web traffic on the Internet. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2014).

[249] QIAN, F., GERBER, A., MAO, Z. M., SEN, S., SPATSCHECK, O., AND WILLINGER, W. TCP revisited: A fresh look at TCP in the wild. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[250] RAAKE, A. Predicting speech quality under random packet loss: Individual impairment and additivity with other network impairments. *Acta Acustia 90* (2004).

[251] RAGHAVAN, S., AND GARCIA-MOLINA, H. Crawling the hidden Web. Stanford InfoLab, 2000.

[252] RASMUSSEN, K., WILSON, A., AND HINDLE, A. Green mining: Energy consumption of advertisement blocking methods. In *Proceedings of the International Workshop on Green and Sustainable Software (GREENS)* (2014).

[253] RICHTER, P., ALLMAN, M., BUSH, R., AND PAXSON, V. A primer on IPv4 scarcity. *ACM SIGCOMM Computer Communication Review (CCR) 45*, 2 (2015).

[254] RICHTER, P., CHATZIS, N., SMARAGDAKIS, G., FELDMANN, A., AND WILLINGER, W. Distilling the Internet's application mix from packet-sampled traffic. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2015).

[255] RIGNEY, C., WILLENS, S., RUBENS, A., AND SIMPSON, W. Remote Authentication Dial In User Service (RADIUS). RFC 2865 (Draft Standard), June 2000. Updated by RFCs 2868, 3575, 5080, 6929.

[256] ROSENBERG, E. The business of Google. http://www.investopedia.com/articles/investing/020515/business-google.asp, 2015. Investopedia.com.

[257] SALOWEY, J., AND DROMS, R. RADIUS Delegated-IPv6-Prefix attribute. RFC 4818 (Proposed Standard), Apr. 2007.

[258] SALTZER, J. H., REED, D. P., AND CLARK, D. D. End-to-end arguments in system design. *ACM Transactions on Computer Systems 2*, 4 (Nov. 1984).

[259] SÁNCHEZ, M. A., OTTO, J. S., BISCHOF, Z. S., CHOFFNES, D. R., BUSTAMANTE, F. E., KRISHNAMURTHY, B., AND WILLINGER, W. Dasu: Pushing experiments to the Internet's edge. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2013).

[260] SÁNCHEZ, M. A., OTTO, J. S., BISCHOF, Z. S., CHOFFNES, D. R., BUSTAMANTE, F. E., KRISHNAMURTHY, B., AND WILLINGER, W. A measurement experimentation platform at the Internet's edge. *IEEE/ACM Transactions on Networking 23*, 6 (Dec 2015).

[261] SARGENT, M., AND ALLMAN, M. Performance within a fiber-to-the-home network. *ACM SIGCOMM Computer Communication Review (CCR) 44*, 3 (July 2014).

[262] SARRAR, N., MAIER, G., AGER, B., SOMMER, R., AND UHLIG, S. Investigating IPv6 traffic. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2012).

[263] SAT, B., AND WAH, B. W. Analyzing voice quality in popular VoIP applications. *IEEE MultiMedia 16*, 1 (2009).

[264] SCHINAZI, D. Apple and IPv6 - Happy eyeballs. https://www.ietf.org/mail-archive/web/v6ops/current/msg22455.html, 2015.

[265] SCHNEIDER, F., AGER, B., MAIER, G., FELDMANN, A., AND UHLIG, S. Pitfalls in HTTP traffic measurements and analysis. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2012).

[266] SEQUEIRA, L., ET AL. The influence of the buffer size in packet loss for competing multimedia and bursty traffic. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)* (2013).

[267] sFlow. http://sflow.org/.

[268] SHADBOLT, N., BERNERS-LEE, T., AND HALL, W. The semantic Web revisited. *IEEE Intelligent Systems 21*, 3 (2006).

[269] SHAFIQ, M. Z., JI, L., LIU, A. X., PANG, J., AND WANG, J. A first look at cellular machine-to-machine traffic – Large scale measurement and characterization. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)* (2012).

[270] SHERRY, J., HASAN, S., SCOTT, C., KRISHNAMURTHY, A., RATSANAMY, S., AND SEKAR, V. Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (2012).

[271] SINGH, H., BEEBEE, W., DONLEY, C., AND STARK, B. Basic requirements for IPv6 customer edge routers. RFC 7084 (Informational), Nov. 2013.

[272] SINGLA, A., CHANDRASEKARAN, B., GODFREY, P. B., AND MAGGS, B. The Internet at the speed of light. In *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)* (2014).

[273] SITARAMAN, R. K., KASBEKAR, M., LICHTENSTEIN, W., AND JAIN, M. *Overlay networks: An Akamai perspective*. John Wiley & Sons, 2014.

[274] SOMMERS, J., BARFORD, P., GREENBERG, A., AND WILLINGER, W. An SLA perspective on the router buffer sizing problem. *ACM SIGMETRICS Performance Evaluation Review (PER) 35*, 4 (2008).

[275] SOMMERS, J., KIM, H., AND BARFORD, P. Harpoon: A flow-level traffic generator for router and network tests. *ACM SIGMETRICS Performance Evaluation Review (PER) 32*, 1 (2004).

[276] SPRINGBORN, K., AND BARFORD, P. Impression fraud in online advertising via pay-per-view networks. In *Proceedings of the USENIX Security Symposium* (2013).

[277] STONE-GROSS, B., STEVENS, R., ZARRAS, A., KEMMERER, R., KRUEGEL, C., AND VIGNA, G. Understanding fraudulent activities in online ad exchanges. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[278] STREIBELT, F., BOETTGER, J., CHATZIS, N., SMARAGDAKIS, G., AND FELDMANN, A. Exploring EDNS-client-subnet adopters in your free time. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[279] STROHMEIER, D., JUMISKO-PYYKKO, S., AND RAAKE, A. Toward task-dependent evaluation of Web-QoE: Free exploration vs. “who ate what?”. In *Proceedings of IEEE Globecom Workshops* (2012).

[280] STROHMEIER, D., MIKKOLA, M., AND RAAKE, A. The importance of task completion times for modeling web-QoE of consecutive web page requests. In *Proceedings of the IEEE International Workshop on Quality of Multimedia Experience (QoMEX)* (2013).

[281] SUN, L. *Speech quality prediction for voice over Internet protocol networks*. PhD thesis, Univ. of Plymouth, 2004.

[282] SUNDARESAN, S., BURNETT, S., FEAMSTER, N., AND DE DONATO, W. BISmark: A testbed for deploying measurements and applications in broadband access networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)* (2014).

[283] SUNDARESAN, S., DE DONATO, W., FEAMSTER, N., TEIXEIRA, R., CRAWFORD, S., AND PESCAPÈ, A. Broadband Internet performance: A view from the gateway. *ACM SIGCOMM Computer Communication Review (CCR) 41*, 4 (2011).

[284] SUNDARESAN, S., DE DONATO, W., FEAMSTER, N., TEIXEIRA, R., CRAWFORD, S., AND PESCAPÈ, A. Measuring home broadband performance. *Communications of the ACM 55*, 11 (2012).

[285] SUNDARESAN, S., FEAMSTER, N., TEIXEIRA, R., AND MAGHAREI, N. Measuring and mitigating Web performance bottlenecks in broadband access networks. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2013).

[286] TANENBAUM, A. S., AND WETHERALL, D. J. *Computer Networks*, 5th ed. Prentice Hall Press, 2010.

[287] THALER, D., DRAVES, R., MATSUMOTO, A., AND CHOWN, T. Default address selection for Internet Protocol version 6 (IPv6). RFC 6724 (Proposed Standard), Sept. 2012.

[288] THELWALL, M., AND STUART, D. Web crawling ethics revisited: Cost, privacy, and denial of service. *Journal of the American Society for Information Science and Technology 57*, 13 (Nov. 2006).

[289] THU, Q. H., AND GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters 44*, 13 (June 2008).

[290] TOUBIANA, V., NARAYANAN, A., BONEH, D., NISSENBAUM, H., AND BAROCAS, S. Adnostic: Privacy preserving targeted advertising. In *Proceedings Network and Distributed System Symposium* (2010).

[291] TRIUKOSE, S., WEN, Z., AND RABINOVICH, M. Measuring a commercial content delivery network. In *Proceedings of the International World Wide Web conference (WWW)* (2011).

[292] VALLINA-RODRÍGUEZ, N., SHAH, J., FINAMORE, A., GRUNENBERGER, Y., PAPAGIANNAKI, K., HADDADI, H., AND CROWCROFT, J. Breaking for commercials: Characterizing mobile advertising. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2012).

[293] VILLAMIZAR, C., AND SONG, C. High performance TCP in ANSNET. *ACM SIGCOMM Computer Communication Review (CCR) 24*, 5 (Oct. 1994).

[294] VISHWANATH, A., SIVARAMAN, V., AND THOTTAN, M. Perspectives on router buffer sizing: Recent results and open problems. *ACM SIGCOMM Computer Communication Review (CCR) 39*, 2 (2009).

[295] WANG, J. A Survey of Web caching schemes for the Internet. *ACM SIGCOMM Computer Communication Review (CCR) 29*, 5 (1999).

[296] WANG, M., AND GANJALI, Y. The effects of fairness in buffer sizing. In *Proceedings of the IFIP conference on Ad Hoc and sensor networks, wireless networks, next generation Internet* (2007).

[297] WANG, X., KRISHNAMURTHY, A., AND WETHERALL, D. Speeding up Web page loads with Shandian. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2016).

[298] WANG, X. S., BALASUBRAMANIAN, A., KRISHNAMURTHY, A., AND WETHERALL, D. Demystifying page load performance with WProf. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2013).

[299] WANG, X. S., BALASUBRAMANIAN, A., KRISHNAMURTHY, A., AND WETHERALL, D. How speedy is SPDY? In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)* (2014).

[300] WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13* (2004).

[301] WANG, Z., LU, L., AND BOVIK, A. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication 19*, 2 (2004).

[302] WEAVER, N., AND GETTYS, J. Bufferbloat: What's wrong with the Internet? *Communications of the ACM 55*, 2 (2012).

[303] WEAVER, N., KREIBICH, C., DAM, M., AND PAXSON, V. Here be Web proxies. In *Proceedings of the International Conference on Passive and Active Measurement (PAM)* (2014).

[304] WING, D., AND YOURTCHENKO, A. Happy eyeballs: Success with dual-stack hosts. RFC 6555 (Proposed standard), Apr. 2012.

[305] XIE, G., ILIOFOTOU, M., KARAGIANNIS, T., FALOUTSOS, M., AND JIN, Y. Resurf: Reconstructing Web-surfing activity from network traffic. In *Proceedings of the IFIP Networking conference (NETWORKING)* (2013).

[306] XING, X., MENG, W., LEE, B., WEINSBERG, U., SHETH, A., PERDISCI, R., AND LEE, W. Understanding malvertising through ad-injecting browser extensions. In *Proceedings of the International World Wide Web conference (WWW)* (2015).

[307] YUAN, S., WANG, J., AND ZHAO, X. Real-time bidding for online advertising: Measurement and analysis. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)* (2013).

[308] ZARRAS, A., KAPRAVELOS, A., STRINGHINI, G., HOLZ, T., KRUEGEL, C., AND VIGNA, G. The dark alleys of Madison avenue: Understanding malicious advertisements. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2014).

[309] ZINNER, T., ABBOUD, O., HOHLFELD, O., HOSSFELD, T., AND TRAN-GIA, P. Towards QoE management for scalable video streaming. In *ITC specialist seminar on multimedia applications - Traffic, performance and QoE* (2010).