A Longitudinal View of Dual-stacked Websites – Failures, Latency and Happy Eyeballs

Vaibhav Bajpai* and Jürgen Schönwälder†
*Technische Universität München, Germany
bajpaiv@in.tum.de
†Jacobs University Bremen, Germany
j.schoenwaelder@jacobs-university.de

***Abstract*—IPv6 measurement studies have focussed on measuring IPv6 adoption, while studies on measuring IPv6 performance have either become dated or only provide a snapshot view. We provide a longitudinal view of performance of dual-stacked websites. We show that (since 2013) latency towards ALEXA 10K websites with AAAA entries over the six years have reduced by 29% over IPv4 and by 57% over IPv6. As of Dec 2018, 56% of these websites are faster over IPv6 with 95% of the rest being at most 1 ms slower. We also identify glitches in web content delivery that once fixed can help improve user experience over IPv6. Using a publicly available dataset, we show that 40% of ALEXA 1M websites with AAAA entries were not accessible over IPv6 in 2009. These complete failures have reduced to 1.9% as of Jan 2019. However, our data collection on partial failures helps identify further that 27% of these popular websites with AAAA entries still suffer from partial failure over IPv6. These partial failures are affected by DNS resolution errors on images, javascript and CSS content. For 12% of these websites, more than half of the content belonging to same-origin sources fails over IPv6, while analytics and third-party advertisements contribute to failures from cross-origin sources. Our results also contribute to the IETF standardisation process. We witness that using an Happy Eyeballs (HE) timer value of 250 ms, clients prefer IPv6 connections to 99% of ALEXA 10K websites (with AAAA entries) more than 96% of the time. Although, this makes clients prefer slower IPv6 connections in 81% of the cases. Our results show that a HE timer value of 150 ms does not severly affect IPv6 preference towards websites. The entire dataset presenting results on partial failures, latency and HE used in this study is publicly released.**

I. Introduction

Early IPv6 measurement studies [1]–[3] (2010-2014) have focussed on measuring IPv6 adoption on the Internet. This involved measuring addressing, naming, routing and reachability aspects of IPv6. Early studies measuring IPv6 performance have become dated [2], [4] (2011-2012) since the IPv6 landscape has changed significantly in recent years. For instance, the fraction of ALEXA 1M websites that used to announce AAAA entries in the DNS in 2012 was around ∼1% and it has increased to ∼19.2% as of Jan 2019 (see Fig. 9). Similarly, Google's IPv6 adoption statistics [5] show that the number of connections to Google over IPv6 was less than 1% in 2012 and has increased to ∼26% as of Jan 2019 (see Fig. 1). A number of events have contributed to the change of the IPv6 landscape. The World IPv6 Day (W6D) in 2011 [6] and the World IPv6 Launch Day (W6LD) in 2012 [7] encouraged several notable content providers to start providing services over both IPv6 and IPv4. The rapidly exhausting pool of IPv4 address space has also been a driving factor. As of Jan 2019, four out of five Regional Internet Registries (RIRs) – APNIC (in Apr 2011), RIPE (in Sep 2012), LACNIC (in June 2014), and ARIN (in Sep 2015) have exhausted their IPv4 address pool [8] and consequently Local Internet Registries (LIRs) now receive allocations from the last available IPv4 /8 address block. As a result, within a span of few years, several large IPv6 rollouts have happened [9] both in fixed-line networks (such as Telenet, Belgacom, VOO in Belgium, Swisscom in Switzerland, Comcast in the US, Deutsche Telekom and Kabel Deutschland in Germany) and cellular networks (such as AT&T, Verizon Wireless and T-mobile USA). Comcast recently completed the transition of their entire broadband network infrastructure to be 100% IPv6 ready [10]. These efforts have eventually led to an increased global adoption of IPv6. According to Google's IPv6 adoption statistics [5], ∼26% of connections to Google are served over IPv6 (as of Jan 2019) with Belgium (∼54%), Germany (∼41%), Greece (∼36%), US (∼34%) and India (∼33%) leading in terms of per country IPv6 adoption rates. Fixed-line service providers such as Comcast and Swisscom estimate IPv6 traffic within their network to be ∼25% of the total traffic [11].

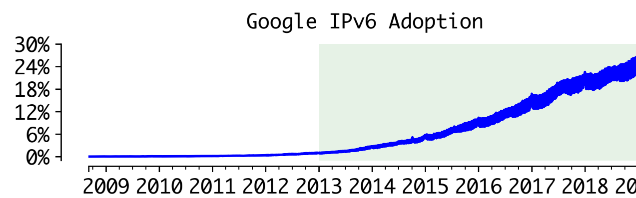

Fig. 1. *Timeseries of fraction of connections reaching Google over IPv6 [5]. The shaded area represents the duration of this study. The dataset is made available by Google at: https://goo.gl/vXqgQK.*

IPv6 carries a noticeable amount of Internet traffic today and it is therefore important to study how IPv6 performs relative to IPv4. Although some recent studies of IPv6 performance do exist [12]–[14], they only provide a snapshot view on the state of IPv6 performance. The only longitudinal work on measuring IPv6 performance is done by Huston [15]. He uses Google Ads to make dual-stacked clients measure latencies towards

APNIC servers. This work is orthogonal to our study since the goal in [15] is to sample a large number of users while we focus on sampling a large number of websites. In fact, Czyz *et al.* [3] (2014) measured round-trip times with 10- and 20- traceroute hop distances and concede that a measure of client-to-service performance would be an ideal metric. Using a six year-long dataset, we measure failures and latency trends from the edge of the network towards operational web content delivery services on the Internet. Our three main findings are summarised as follows:

1. Failures: Using a publicly available dataset, we find that 40% of ALEXA 1M websites with `AAAA` entries used to fail completely over IPv6 in 2009. These failures have reduced to ∼1.9% as of Jan 2019. We observe that (see § IV) 88% of failing websites fall below ALEXA rank 100K, while 1% of failing websites fall above ALEXA rank 10K. In order to measure the extent of partial failures over IPv6, we developed `simweb`, a tool that can download a root webpage and all its referenced webpage elements one-level deep. Using the collected dataset, we find that 27% of ALEXA 100 websites with `AAAA` entries fail partially over IPv6, with 9% having more than 50% partial failures over IPv6. These partial failures are affected by DNS resolution errors on images, javascript and CSS content (see § V). For 12% of these websites, more than 50% of the content belonging to same-origin source fails over IPv6. Content failing from cross-origin sources consists of analytics and third-party advertisements.

Complete and partial failures silently exist since users do not notice them as long as the content can still be accessed over IPv4. These observations raise the question whether a website with partial failures over IPv6 can be considered IPv6-ready.

2. Latency: In order to measure latency over IPv6, we developed `happy`, a tool that can measure TCP connect times. Using the collected dataset, we find that TCP connect times to ALEXA 10K websites with `AAAA` entries have reduced by 29% over IPv4 and by 57% over IPv6 in the past six years. As of Dec 2018, 56% of the dual-stacked websites with ALEXA rank above 10K are faster over IPv6 (see § VI) with 95% of the rest being at most 1 ms slower and 1% of the websites are 40 ms or more slower over IPv6. This is due to the increased presence of CDN deployments, led by Cloudflare (∼35%), Google (∼16%) and Akamai (∼9%) that contribute to content delivery of more than half of ALEXA top 10K websites with `AAAA` entries over both address families.

3. Happy Eyeballs: HE [16], [17] is a mechanism that gives TCP connection requests over IPv6 a 250 ms advantage to connection requests over IPv4. Our measurements indicate that clients prefer IPv6 connections to 99% of the ALEXA 10K websites with `AAAA` entries (see § VII) more than 96% of the time, even though IPv6 connections are slower in 81% of the cases. Our results help provide a concretisation of the HE timer value to 150 ms (as opposed to a recommended range as proposed within the IETF) which does not severly affect IPv6 preference towards these websites.

This paper builds on results published previously in [18]–[20] but expands the work by covering a longer timespan, by integrating results, by adding new results (see § IV), by adding an extended discussion of related work, and by discussing limitations, lessons learned and implications of this measurement study. The entire six year-long dataset on partial failures, latency and happy eyeballs is made publicly available [21] to the research community. This study is relevant for network operators that are planning or are in early stages of IPv6 deployment. It also provides content providers insights on how to ensure that their content delivery over IPv6 compares to the service delivered over IPv4. We identify glitches in content delivery that once fixed can help improve user experience over IPv6. Finally, our results provide a concretisation of the HE timer value (as opposed to a recommended range) for the IPv6 operations community within the IETF.

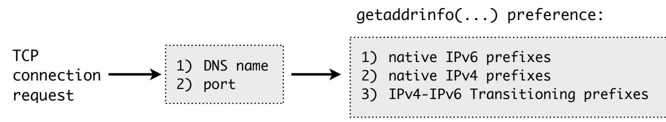

Fig. 2. `getaddrinfo()` *makes applications iterate over endpoints in an order that prefers an IPv6-upgrade path according to RFC 6724.*

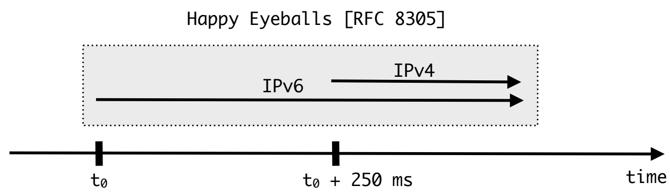

Fig. 3. *The Happy Eyeballs (HE) algorithm that gives a 250 ms advantage to a TCP connection request over IPv6.*

II. Background and Related Work

We begin by describing the behavior of a dual-stacked host. A dual-stacked host with native IPv6 connectivity establishing a TCP connection to a dual-stacked website will prefer IPv6. This is due to the function `getaddrinfo()` that resolves a dual-stacked website to a list of endpoints in an order that prefers an IPv6 upgrade path [22] (2012) as shown in Fig. 2. The dictated order can dramatically reduce the application's responsiveness in situations where IPv6 connectivity is broken. In fact, an attempt to connect over an IPv4 endpoint will only take place when the IPv6 connection attempt has timed out, which can be in the order of several seconds [23].

A. Happy Eyeballs

This degraded user experience can be overcome by implementing the HE algorithm [16], [17] (2017) in applications. The HE algorithm recommends that a host, after resolving the DNS name of a dual-stacked website, tries a TCP `connect()` to the first endpoint (usually IPv6, see Fig. 2). However, instead of waiting for a timeout, which is typically in the order of seconds, it only waits for 250 ms, after which it must initiate another TCP `connect()` to an endpoint with a different address family and start a competition to pick the TCP connection that completes first.

TABLE I. USAGE SHARE OF DESKTOP BROWSERS – JAN 2019

.

Browser	[24]	[25]	[26]
Google Chrome	64%	62%	64%
Mozilla Firefox	14%	6%	10%
Opera	–	3%	2%
Apple Safari	6%	14%	4%

The HE algorithm biases its path selection in favor of IPv6 by design as shown in Fig. 3. It is therefore not designed to encourage aggressive connection requests over IPv4 and IPv6, but instead to satisfy the following goals – (a) The connection requests must be made in an order that honors the destination-address selection policy [22], unless overridden by user or network configuration. The client must prefer IPv6 over IPv4 whenever the policy is not known, (b) The connection initiation must quickly fallback to IPv4 to reduce the wait times for a dual-stacked host in situations where the IPv6 path is broken, and (c) The network path and destination servers must not be thrashed by mere doubling of traffic by making simultaneous connection requests over IPv4 and IPv6. The connection requests over IPv6 must be given a fair chance to succeed before a connection over IPv4 is attempted.

Browser Implementations – Table I shows the usage share of desktop browsers with a HE implementation as of Jan 2019. For instance, Google Chrome has an implementation of the HE algorithm since v11.0.696.71 [27], which was released in 2011. It uses a 300 ms timer [17], which is started after the first TCP SYN request over IPv6 has been sent. Once the timer expires (300 ms) the browser switches to IPv4 and starts a competition between IPv4 and IPv6 connection requests to pick the one that completes first. Mozilla Firefox released its first HE implementation with v7.0. The implementation received multiple bug reports leading to a stable implementation by v15.0 [28]. Firefox by default, unlike Google Chrome follows a more aggressive approach by starting parallel TCP connections to the first endpoints of each address family. However, once one of the connections has been successfully established, the second connection request is not closed by sending a TCP RST, instead the connection request is allowed to continue until the request times out. Opera, since v12.10 [29] has an implementation similar to that of Mozilla Firefox. It tries simultaneous TCP connections to the first endpoint of each address family and chooses whichever completes first. It remains unclear whether parallel connection attempts can be deemed as a flavor of HE, since the algorithm is designed to honor the IPv6 upgrade policy and therefore does not encourage aggressive connection requests over IPv4 and IPv6. As such, Mozilla Firefox also allows to disable parallel connection attempts by setting a parameter, `network.http.fast-fallback-to-IPv4` to `false`, after which the browser starts preferring IPv6 connection requests with a 250 ms timer value. Apple Safari prior to OS X 10.11 (since OS X 10.7) [30] used a more hybrid approach. The OS X networking APIs maintained a history of the previously witnessed latencies to each destination along with a combined mean for each address family. Apple Safari instead of using `getaddrinfo()` used these higher level APIs to prefer the fastest connection. Moreover, Apple Safari did not switch to a different address family if no response was received from the first endpoint, instead it tried a TCP connection with the next endpoint in the same address family. This took a long time for an address family switch-over. Apple with OS X 10.11 and iOS 9 has a new simplified HE implementation [31] which uses a 25 ms timer value in favour of IPv6 connections. These HE timer values are arbitrarily chosen. Using the longitudinal dataset, we study the effects (see § VII for details) of lowering the HE timer value.

B. *Related Work*

Measuring IPv6 Adoption – Zander *et al.* [32], [33] (2012) use Google ads to measure IPv6 client capabilities. Karir *et al.* [34] (2013) use this technique to study the amount and nature of IPv6 population based on location, type of transition technology and ISP. Dhamdhere *et al.* [2] (2012) study IPv6 topology and routing dynamics and provide comparisons with IPv4. Czyz *et al.* [3] (2014) study IPv6 adoption based on addressing, naming, routing and end-to-end reachability aspects and compare dual-stack usage profiles. Berger *et al.* [35] (2013) present a passive technique to pair addresses of dual-stacked DNS resolvers. They take this further [36] (2015) and use TCP-based fingerprinting to identify IP endpoints that belong to same server machines. Giotsas *et al.* [37] (2015) study the congruity between AS topologies and reason Hurricane Electric to be the primary contributor of the disparity in IPv6. Nikkhah *et al.* [38] (2016) propose a model to capture the decisions of stakeholders and show that with limited coordination, the parameters of the model affecting IPv6 migration become predictable. Recently there has also been an increased interest in efficiently scanning the IPv6 address space [39]–[43], on techniques that help uncover spatio/temporal [44]–[46] structure of IPv6 addresses and security-related [47], [48] aspects of IPv6.

Measuring IPv6 Performance – Colitti *et al.* [1] (2010) measure latency using HTTP requests to two experimental Google web service hostnames using a fraction of Google users. The methodology described in this work became the basis for the Google IPv6 adoption statistics as shown in Fig. 1. Nikkhah *et al.* [4] (2011) measure average download speeds towards ALEXA 1M websites from 6 vantage points. Dhamdhere *et al.* [2] (2012) measure page load times towards ALEXA 1M websites from 5 vantage points. Both [4] and [2] show that IPv6 performance is comparable to IPv4 when forward AS-level paths are same, but much worse when they differ. Dhamdhere *et al.* [2] also show that page fetch times (due to small size of typical pages) are more dominated by delay rather than available bandwidth. This is why we use TCP connect times as a metric for measuring performance of dual-stacked websites (see § VI) since it allows us to capture this end-to-end delay at the transport layer. Alzoubi *et al.* [49] (2013) study performance implications of unilateral enabling of services over IPv6. They witnessed no performance penalty in disabling the opt-in service. As can be seen these studies

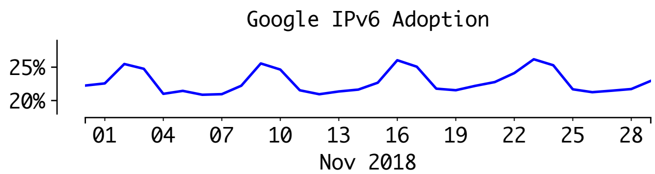

Fig. 4. *IPv6 adoption trends follow a weekly pattern with more connections using IPv6 to reach Google over the weekends. IPv6 penetration is more in residential deployments than in corporate networks. The dataset is made available by Google at: https://goo.gl/vXqgQK.*

are dated. Recently, Livadariu *et al.* [12] (2016) study the stability of IPv6 in the control and data plane. This study is orthogonal and complements our work on measuring TCP connect times. Goel *et al.* [13] (2016) measure RTT, DNS lookup, and page load times as seen by the Akamai monitoring system and show that IPv6 performs better than IPv4 in US cellular networks. Pujol *et al.* [14] (2017) study DNS and flow-level statistics collected from an ISP. They show that ∼80% of the RTT observed in the backbone over IPv6 are 10 ms apart from IPv4. Using a `youtube` [50] test, we recently [51] (2017) measured content delivery towards Youtube media server destinations and observed consistently higher TCP connection establishment times and startup delays over IPv6. On further investigation, we observed [52] (2018) that latency over both address families was comparable in situations where content caches were dual-stacked. We found that these cache deployments shorten IP path lengths by roughly up to 50% and latencies are reduced by up to 33% over IPv4 and are halved over IPv6.

Measuring HE – Studies [53]–[55] (2011-2012) in the past have also analysed HE implementations in Mozilla Firefox 7 and 8, Google Chrome 11, Opera 11 and Apple Safari on OS X 10.7. It was witnessed that Google Chrome (with a 300 ms timer) helps reduce the degraded user experience in situations where IPv6 connectivity of the client is broken. Mozilla Firefox (with the fast-fallback parameter disabled) has an HE behaviour similar to that of Google Chrome. Apple Safari on OS X 10.7 tends to prefer the fastest connection, but in the process also prefers legacy IPv4 connectivity even where IPv6 connectivity is relatively similar, a situation referred to as *hampering eyeballs*, since it tends to delay the transition to IPv6. These studies however are again dated since HE behavior in browser implementations has changed with time.

Baker [56] (2012) describes HE metrics and testbed configurations in a controlled setting to measure how quickly an application can reliably establish connections from a dual-stacked environment. Zander *et al.* [32] (2012) showed that 20% of the hosts had a HE implementation, out of which 75% of the connection attempts preferred IPv6. We show that this preference (see § VI for details), due to decreased latencies over IPv6 has increased to 96%. They observed that HE was used by hosts running Google Chrome (9% of connections), Apple Safari (4%) and Mozilla Firefox (1%).

As can be seen, there has been more focus on IPv6 adoption, while studies on measuring IPv6 performance have become either dated or provide a snapshot view on the state of IPv6 performance. This is the first study to provide a longitudinal view (six years) of failures and performance of dual-stacked websites as seen from multiple vantage points.

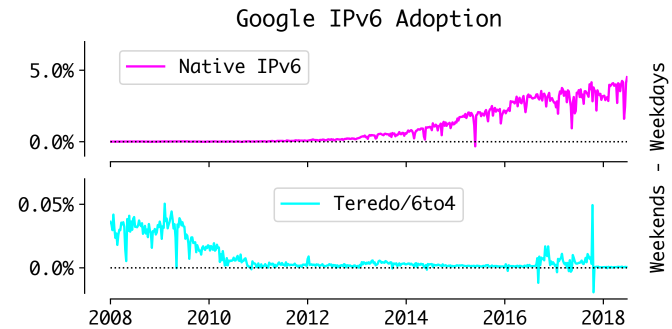

Fig. 5. *The drift (weekends - weekdays) of IPv6 penetration as seen by Google. A positive trend means higher IPv6 adoption over weekends than weekdays. The positive trend for transitioning technologies stopped at around 2011, and started increasing for native IPv6 since 2013. The dataset is made available by Google at: https://goo.gl/vXqgQK.*

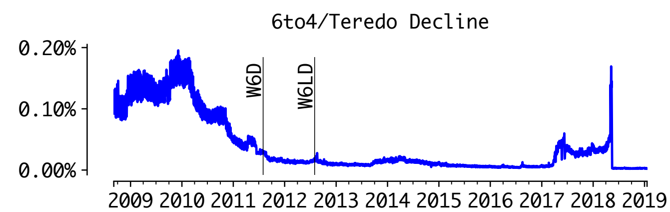

Fig. 6. *Timeseries of fraction of connections using Teredo/6to4 relays to reach Google over IPv6 [5]. These relays add considerable latency over IPv6. The dataset is made available by Google at: https://goo.gl/vXqgQK.*

III. Measurement Platform

We investigated potential measurement platforms that we could leverage for measuring IPv6 performance. For instance, RIPE Atlas [57], [58] with ∼9.8K (around ∼2.4K in Jan 2013 when we began the measurements) connected probes with ∼2.8K dual-stacked probes [59] as of Jan 2019 is ideal, but is limited in the number of metrics it can measure (primarily `ping` and `traceroute`). PlanetLab [60] would be another choice (although IPv6 support was only recently added), but the vantage points are restricted to mostly research networks while we are interested in measuring from different types of networks and particularly residential networks. This is because IPv6 adoption over the years has seen higher penetration in residential deployments than in corporate networks. For instance, Fig. 4 shows that the Google IPv6 adoption trend [5] exhibits a weekly pattern (also witnessed by other CDN [61] providers) with higher IPv6 penetration over the weekends. This shift (weekends vs weekdays) in trend has been steadily increasing with time as shown in Fig. 5, with a parallel decline in the use of transitioning technologies. This indicates that service providers gradually turned off (see Fig. 6) transitioning

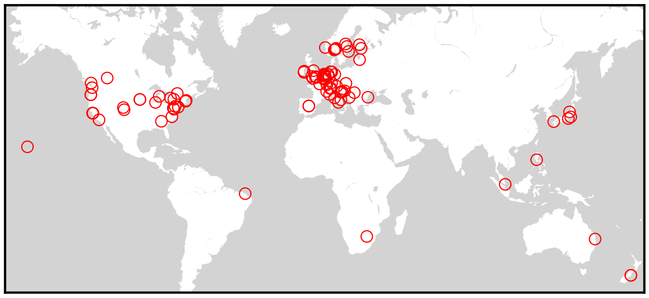

Fig. 7. *Measurement trial of ∼100 dual-stacked SamKnows probes as of Jan 2019. The metadata for each probe is available online: https://goo.gl/NN81ij*

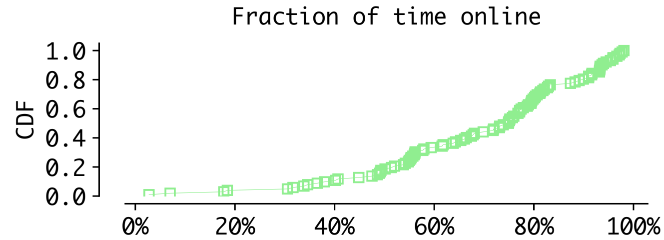

Fig. 8. *Distribution of the fraction of the time each probe is online during its deployment. 75% of the probes are available more than half of the time.*

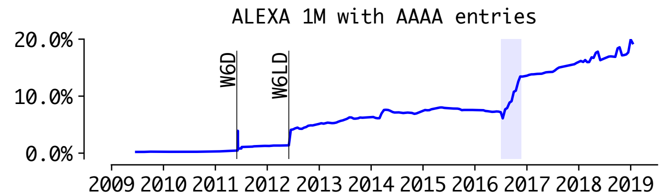

Fig. 9. *Timeseries shows the percentage of ALEXA 1M with AAAA entries [63]. Vertical markers indicate the W6D and W6LD events. The shaded region shows Cloudflare adding AAAA entries to all websites hosted on its CDN infrastructure. The dataset is available online: http://www.employees.org/∼dwing/aaaa-stats*

technologies (around 2011) and started (around 2013) deploying native IPv6 support for home users. Note, the apparent gradual increase in Teredo adoption in 2017 associated with a spike and sudden dropout in 2018 is a measurement error as reported by Google [62]. This adoption has been happening faster in residential networks than in corporate networks. This is the reason why we prefer measuring IPv6 performance from residential settings. BISmark [58] probes are deployed in residential settings, but it is currently unknown how many probes are dual-stacked.

As such, we deployed SamKnows probes at locations with native IPv6 connectivity. SamKnows [58] is a company specializing in the deployment of hardware-based probes that perform continuous measurements to assess broadband performance. SamKnows probes are also used by the FCC as part of the Measuring Broadband America (MBA) project. Fig. 7 shows the current deployment status of ∼100 dual-stacked SamKnows probes representing 66 different origin ASes that are part of this measurement study. Most of the SamKnows probes we deployed are connected within the RIPE (60 probes) and ARIN (29) region and are largely hosted in home networks (78) with native IPv6 connectivity. As such, the observations are biased by the number and location of SamKnows probes which largely cover US, EU and JP regions. Note, a large fraction of IPv6 deployment is also centered in these regions, but we concur that the state of IPv6 adoption may change in the future. In the beginning of the measurement campaign (2013), it was a challenge to recruit volunteers who received native IPv6 connectivity at home (see Fig. 1) and were not only willing to host a probe for us but also keep it running in the long term. In the beginning of 2015, with the deployment of 24 probes and experience gained from managing the probes for multiple months, we ran a dedicated campaign to recruit more volunteers on network operations mailing lists which helped increase the footprint to 81 probes. In late 2016, we ran a smaller second campaign which increased the deployment a bit further by ∼20% leading to 102 SamKnows probes that overall contribute to this study. Fig. 8 shows the fraction of time a probe is online during its lifetime. It can be seen that ∼75% of the probes are available more than half of the time, but a significant fraction of probes also experience downtimes.

A. Measurement Setup

We implemented two measurement tests for this study. The `simweb` test (see § V for details) is used to measure partial failures over IPv6, while the `happy` test (see § VI for details) is used to measure latency over IPv6. We cross-compiled these tests for the OpenWrt platform and deployed it on SamKnows probes. These probes (in addition to the `simweb` and `happy` test), also perform standard SamKnows IPv4 measurements. The measurement tests are open-sourced and publicly released. In pursuit to identify targets for the measurement tests, we investigated the number of websites that provide `AAAA` entries in DNS. Fig. 9 shows the evolution of the number of websites with `AAAA` entries within the top ALEXA 1M websites. The trend shows a peak jump during the W6D (2011) [6], W6LD (2012) [7] and between Jul-Nov 2016. The steady increase (from ∼7% to ∼13%) between Jul-Nov 2016 which was much more significant than W6D (from ∼0.4% to ∼0.8%) and W6LD (from ∼1% to ∼4%) can be attributed to Cloudflare when it decided to remove its opt-in service and add `AAAA` entries for all websites [64] hosted on its CDN infrastructure by default. The distribution shows that ∼192K websites announce `AAAA` entries in DNS as of Jan 2019. We prepend each website name with the label `www` (since some websites provide `AAAA` records only for domain names starting with `www`) to make an additional DNS request and we also explicitly follow `CNAME`s. The measurement tests run once every hour. Due to the inherent storage limitation of the probes, the locally collected measurement results are pushed every hour to our data collector over HTTPS as shown in Fig. 10. For latency measurements, we limit probing to top 10K websites since less than 1% of websites (see § IV) that

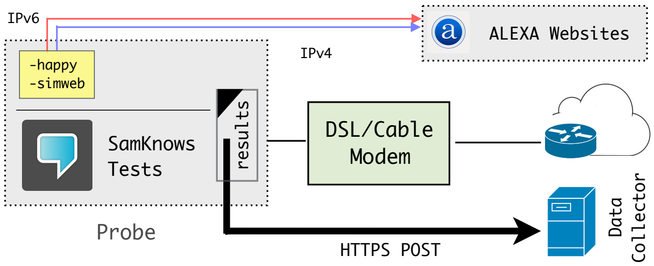

Fig. 10. *A measurement setup on top of the SamKnows platform. A dual-stacked probe, in addition to the standard SamKnows tests, executes happy and simweb tests. These tests run every hour and measure towards ALEXA websites both over IPv4 and IPv6. The locally collected measurement results are pushed every hour to a data collector using HTTP.*

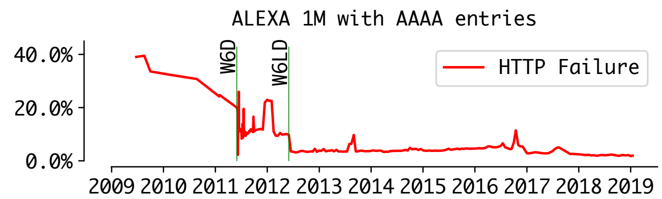

Fig. 11. *Timeseries shows the percentage of HTTP failure rates witnessed when accessing ALEXA 1M websites with AAAA entries over IPv6 [63]. Spikes correspond to the W6D (2011) and W6LD (2012) events. The dataset is available online: http://www.employees.org/~dwing/aaaa-stats*

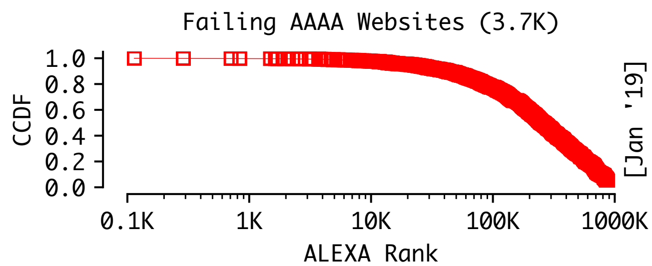

Fig. 12. *CCDF of ALEXA ranks of AAAA websites that fail to establish a HTTP session over IPv6 as of Jan 2019. 1% of these failing websites have ranks above 10K with ~48 websites above rank 10K. 88% of failing websites have ranks below 100K.*

announce AAAA entries in DNS and fail to establish a HTTP session over IPv6 have ranks above 10K. Furthermore, since probes are deployed at home and measurements repeat every hour, we do not want to overwhelm the volunteer's broadband connection with our measurement traffic. The `happy` dataset is ~794 GB and the `simweb` dataset is ~754 GB, contributing to an overall disk space utilisation of ~1.5 TB.

IV. Complete Failures

Content providers need to ensure that the content delivered over IPv4 and IPv6 is identical. This is a 2-step process, whereby the content provider has to begin by providing an `AAAA` record of the service endpoint (or the upfront load balancer) to the DNS resolvers. The end-host then must be able to receive the same content when requesting services from the resolved IPv6 endpoint. IPv6 adoption studies (see § II) have mostly focussed on the first step by measuring the amount of `AAAA` entries in DNS. We take this further by quantifying the failure rates of websites that provide `AAAA` entries in DNS. In this section, using a publicly available dataset, we investigate complete failures whereby either the TCP connection to the website times out or the HTTP session to the root webpage fails to complete succesfully. Fig. 11 shows the timeseries of HTTP failures of dual-stacked websites within the ALEXA 1M list. These websites announce `AAAA` entries in DNS but fail to establish a HTTP session over IPv6. It can be seen that in 2009, ~40% (out of ~2K) `AAAA` websites used to fail when establishing an HTTP session over IPv6. These failure rates halved (~20%, out of ~4.3K `AAAA` websites) by the W6D (2011) and in one year, further reduced to ~10% (out of ~13K `AAAA` websites) by the W6LD (2012). As of Jan 2019, ~1.9% (out of ~192K) `AAAA` websites fail to establish a HTTP session over IPv6.

We further investigated the ALEXA rankings of these `AAAA` websites that fail over HTTP. Fig. 12 shows the distribution of ranking of failing (~1.9%) `AAAA` websites. As of Jan 2019, ~3.7K (out of ~192K) `AAAA` websites fail to establish a HTTP session over IPv6 with ~1% of these failing websites having ranks above 10K. This includes ~48 websites above rank 10K. However, the distribution is heavily tailed with ~88% of failing websites having ranks below 100K with ~64% having ranks below 300K.

We also investigated whether websites stopped providing IPv6 support over the duration of our measurement campaign. We found five websites that were accessible over IPv6 in the past (since 2013), but have stopped providing `AAAA` entries and exhibit complete failure over IPv6. For instance, `www.bing.com` permanently stopped (even though `www.microsoft.com` and `www.office.com` are still IPv6 enabled) providing IPv6 in Sep 2013. We do not remove such websites from our target list, since we expect them to announce `AAAA` entries in the foreseeable future. For instance, `www.engadget.com` is hosted on Amazon EC2, who starting Jan 2017 added IPv6 support for all AWS regions and we expect the website to resume IPv6 services soon.

Summary — Number of ALEXA 1M websites with AAAA entries that consistently fail to establish an HTTP session over IPv6 have reduced from 40% (2009) to 1.9% (2019). 88% of websites with AAAA entries have ALEXA ranks below 100K that exhibit consistent failure to establish an HTTP session over IPv6 as of Jan 2019.

V. Partial Failures

In situations where the HTTP session to the root webpage is established successfully over both address families, the fraction of the content that can be fetched over IPv6 (without IPv4 fallback) needs further investigation. For instance, previous work [2], [4] has filtered out dual-stacked websites where content over IPv4 and IPv6 was found to be not within a certain

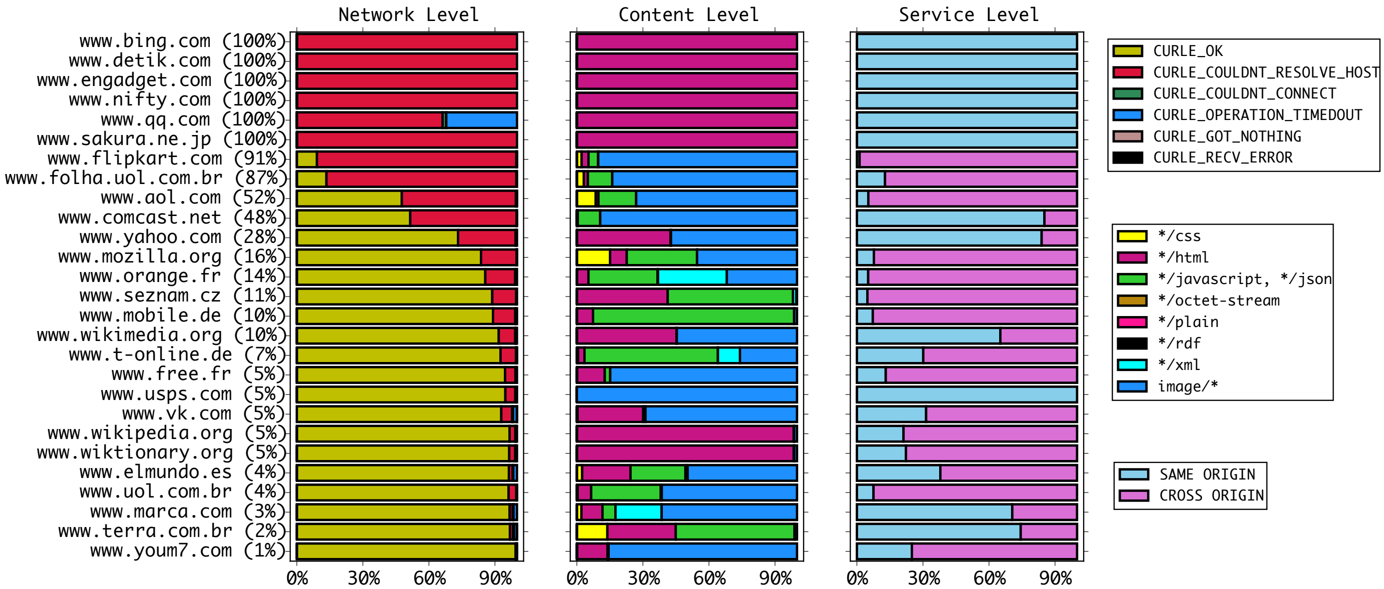

Fig. 13. *The causal analysis of websites failing over IPv6 at the network (left), content (middle) and service level (right). The percentage next to each website is its failure rate. Failure rates are affected by DNS resolution error on images, javascript and CSS content delivered from both same- and cross-origin sources. The description of the* `curl` *error codes is available at: https://curl.haxx.se/libcurl/c/libcurl-errors.html. The raw values are available at: https://goo.gl/5Pt1DF.*

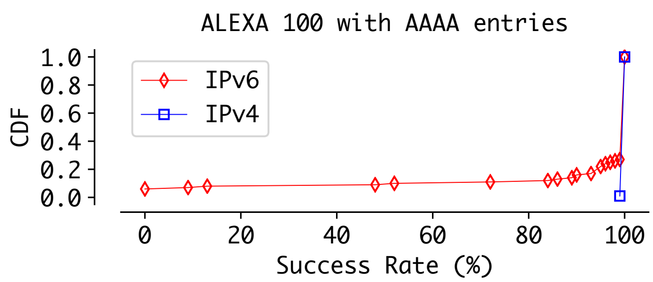

Fig. 14. *Distribution of success rates towards ALEXA top 100 websites with AAAA entries. 27% of websites have fraction of webpage elements that fail over IPv6 with 9% having more than 50% webpage elements that fail over IPv6. The list of websites with partial failures is available at: https://goo.gl/9uVTWD.*

threshold of each other. As a consequence, no further analysis on partial failures was performed. We close this gap. We want to know, in what percentage of cases a website accessible over IPv6 is ostensibly dependent on IPv4 content? What fraction of the content of such a website can be fetched over IPv6 alone? In situations where the entire content cannot be fetched over IPv6 alone, what factors contribute towards IPv4 fallback?

In order to address this question, we implement `simweb` (`https://github.com/steffiejacob/simweb`), a test written in C that we use to measure partial failures over IPv4 and IPv6. For a given website, `simweb` downloads the root webpage and all its referenced webpage elements one-level deep only. In the process it calculates the content type, content size, resource URL, and IP endpoint used to fetch each webpage element. These properties are reported both over IPv4 and IPv6. In addition, HTTP status codes and `curl` response codes are also used to identify the network level status of each request. The reported content size is the size of the payload (excluding the header). In situations where the response is HTTP chunked encoded [65], the payload is the sum of the size of all chunks (excluding the chunked metadata). In situations where the response is compressed, the content size reports the payload size before the receiver decompresses the data. The partial failures are calculated in the data analysis phase using well-known content and service complexity metrics [66], [67]. In situations where there are partial failures we perform a causal analysis and identify sources responsible for the failures.

We witnessed that 27% of dual-stacked websites have some fraction of webpage elements that fail over IPv6 with 9% of the websites having more than 50% webpage elements that fail over IPv6 as shown in Fig. 14.

We further performed a causal analysis to investigate the network, content and service level source of these failing elements. Fig. 13 (left) shows the percentage contribution of `libcurl` error codes to each failing website. The numbers next to each failing website are the failure rates. The error code `CURLE_OK` contributes to the success rate, while rest of the error codes contribute to the failure rate of each website over IPv6. It can be seen that `CURLE_COULDNT_RESOLVE_HOST` is the major contributor to failure rates. This shows that most of the webpage elements fail due to a DNS resolution error. This is caused due to missing `AAAA` entries for these webpage elements in the DNS.

We also investigated MIME types reported by `simweb` for each object of a failing website. Fig. 13 (middle) shows the percentage contribution of MIME types to each failing

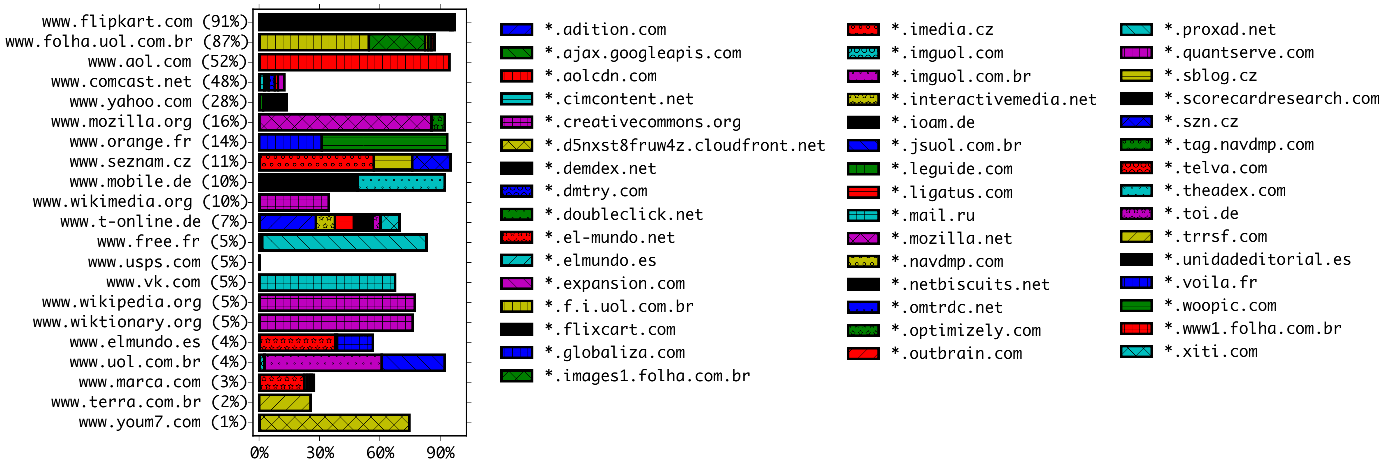

Fig. 15. *Contribution of cross origin sources towards the failure of webpage elements over IPv6. The percentage next to each website is its failure rate. These cross-origin sources can be largely classified as third-party advertisements, analytics, user-centric and static content that tends to fail over IPv6.*

website. It can be seen that images, javascripts, and CSS content contributes to the partial failure of a website over IPv6.

We also investigated the URLs reported by `simweb` for each webpage element that fails over IPv6. We used URLs to classify elements into *same origin* and *cross origin* sources. We classify objects of a website to belong to a same origin source whenever their hostnames match the second level domain of the website. Fig. 13 (right) shows that websites with partial failure have some webpage elements that belong to the same origin source and fail over IPv6. Worse, 12% of these websites have more than 50% webpage elements that belong to the same origin source and fail over IPv6. This is because the CDN that serves the content of a website does not have IPv6 turned on by default for all same-origin webpage elements.

Fig. 15 shows the contribution of webpage elements that belong to cross origin sources. These cross-origin sources can be largely classified [68] as third-party advertisements, analytics, user-centric content, and static content that tends to fail over IPv6. Given some of the cross-origin sources contribute to the failure of multiple websites, we also identified cross-origin sources that would help benefit more websites if their content was available over IPv6. For instance, cross-origin source `doubleclick.net` has the highest span across 5 websites, with a 0.54% median contribution to failure rates. The cross-origin source `creativecommons.org` on the other hand has 76% median contribution to the failure rate of 3 websites. The number of spanned websites show how many websites can be benefited, with the median contribution exhibiting how much the failure rate can be reduced by enabling IPv6 for these cross-origin sources.

A webpage is allowed to embed elements from cross-origin sources that may not provide delivery over IPv6. As such, this is not a violation of expected behavior. However, such partial failures tend to silently exist since dual-stacked clients fallback to IPv4 to fetch those webpage elements. We hope quantification of such failure cases can help content providers pick and enable IPv6 delivery for such webpage elements.

Summary — *Measurements towards the root webpage of a website alone, can lead to an overestimation of IPv6 adoption numbers. We witnessed that 27% of ALEXA top 100 websites with AAAA entries have images, javascript and CSS content that fails delivery over IPv6, which raises the question whether websites with such partial failures can be considered IPv6-ready. We noticed that the CDN infrastructure did not have IPv6 turned on by default for half of the same-origin content for 12% of ALEXA top 100 websites with AAAA entries that exhibit partial failure. We also observed that cross-origin sources such as `creativecommons.org` contribute to the partial failure of more than one website. These failures silently exist because clients usually do not notice them as long as content is available over IPv4.*

VI. Latency

In situations where webpage elements can be fetched over both address families, the performance aspects of retrieving the content over IPv4 and IPv6 needs an investigation. This is particularly important because the default address selection policy [22] makes clients prefer (see § II for details) connections made over IPv6. However, it is unclear whether users experience benefit (or an added penalty) when connecting to websites over IPv6. In order to address this question, we use a metric that measures the time taken to establish a TCP connection to a given endpoint. The input parameter of the metric is a tuple (service name, port number) and the output is the TCP connection establishment time (typically measured in microseconds) for all endpoints the service name resolves to. The `happy` test (`https://github.com/vbajpai/happy`) is an implementation of the metric. It can read one or more service names at once and apply `getaddrinfo()` to resolve DNS entries to `A` and `AAAA` resource records. It then uses non-blocking TCP `connect()` calls to concurrently establish connections to all endpoints seen in the resource records of each service name. It calculates the time it takes for the TCP

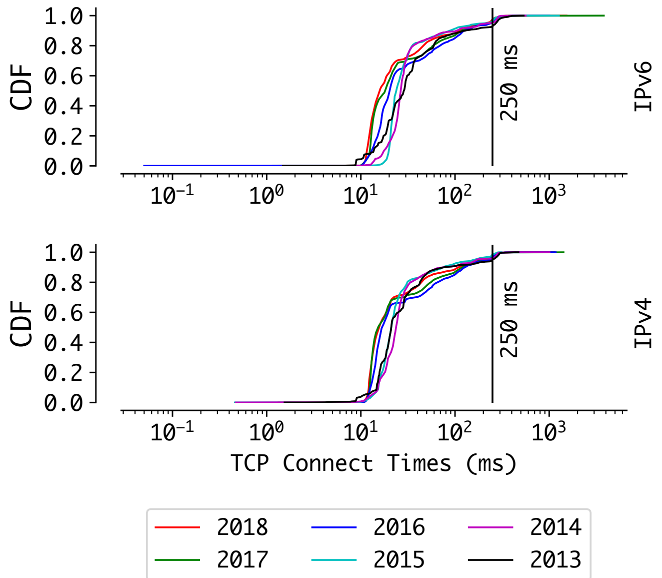

Fig. 16. *CDF of TCP connect times (1M data points) over IPv4 and IPv6 over the six year long duration split by year. The median TCP connect times have reduced in six years by 29% over IPv4 and by 57% over IPv6.*

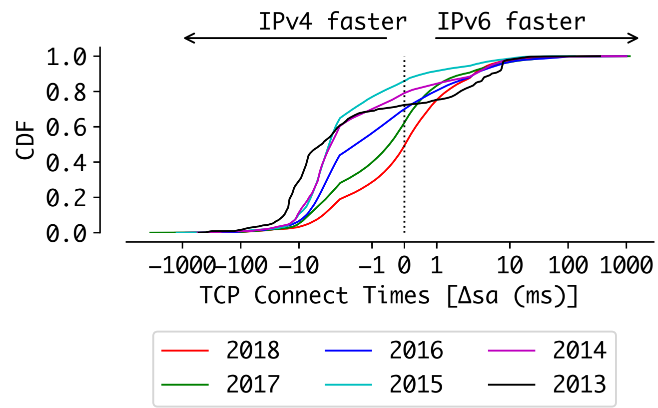

Fig. 17. *CDF of difference in TCP connect times (1M data points) over IPv4 and IPv6 over the six year long duration split by year. Half of the TCP connections that used to be 5 ms or more slower over IPv6 (2013) have reduced to 8% connections as of 2018.*

`connect()` call to complete as a measure of the elapsed time. In order to allow delineating connection timeouts `happy` also keeps a flag as an indication on whether the connection got established. This indication is made once a socket in a `select()` call becomes writeable with no pending socket errors. We do not account the DNS resolution time in the measured connection establishment time. This is done to avoid slow resolvers from biasing our connection establishment time results. The `happy` test enforces a small delay (25 ms by default) between concurrent TCP `connect()` calls to avoid generating bursty `SYN` traffic. This delay, however, does not come in the way of pending TCP `connect()` calls. As such the measured times are not skewed by this feature. We also added the capability to lock the output stream to allow multiple processes to coordinate writes to the same output stream. This is useful when multiple `happy` instances try to append results to a single regular file from a resource-constrained device. By repeated execution of `happy`, we are able to collect time series of TCP connect times that provide us with insights on how IPv6 connectivity to websites compares to that of IPv4.

Fig. 16 shows the distribution of TCP connect times over IPv4 and IPv6 over the duration of the dataset split by year. For instance, back in 2013, TCP connect times were within ∼21 ms (over IPv4) and ∼28 ms (over IPv6) for half of the connections. During the course of six years, the TCP connect times are reduced to ∼15 ms (over IPv4) and ∼16 ms (over IPv6) for half of the connections as seen in 2018. We also paired each connection over IPv4 and IPv6 and calculated the difference in TCP connect times as shown in Fig. 17. The difference $\Delta s_a(u)$ is calculated as $t_4(u) - t_6(u)$, where $t(u)$ is the TCP connect time towards a website u. It can be seen that 72% of the TCP connections used to be (2013) slower over IPv6 with half of the connections being 5 ms or more slower over IPv6. The difference in TCP connect times have reduced over the years whereby half of the TCP connections are slower over IPv6 (2018) with 8% of the connections being 5 ms or more slower over IPv6. Fig. 18 shows time series of difference in TCP connect times towards popular websites. Each data point is a median TCP connect time across all SamKnows probes. This is to ensure that observations do not get biased by a specific SamKnows probe. Note, observations from all google and blogspot websites are grouped together as `www.google.*` and `www.blogspot.*` since they are served by the the same CDN infrastructure (`AS15169`) and therefore tend to offer similar performance over each address family. In fact, a large fraction of these websites are served by CDN deployments. We investigated the percentage of websites above rank ALEXA 10K with AAAA entries that are hosted by a CDN provider. We observed that ∼66% of these websites are served by CDN deployments over IPv4 with a slightly reduced presence (∼60%) over IPv6. Cloudflare (∼35%), Google (∼16%) and Akamai (∼9%) are the leading CDN deployments that contribute towards this CDN penetration. For instance, `www.att.com` (a DSL network provider), `www.comcast.com` (a cable network provider), and `www.irs.gov` (the US tax collection agency) show very similar performance because the websites are served by the Akamai CDN infrastructure. Fig. 18 shows that TCP connect times to popular websites were worse over IPv6 back in 2013 and that the situation has changed with time. It can be seen that TCP connect times to popular websites over IPv6 have improved over time. In fact, Fig. 19 shows that as of Dec 2018, ∼56% of dual-stacked websites with ALEXA rank above 10K are faster over IPv6 with ∼95% of the rest at most 1 ms slower with ∼2% being 25 ms or more slower and ∼1% being 40 ms or more slower over IPv6. On the other hand, ∼8% of the websites are 1 ms or more faster and ∼2% are 10 ms or more faster over IPv6. Facebook recently showed that their news feeds load 30% faster over IPv6 from a US mobile

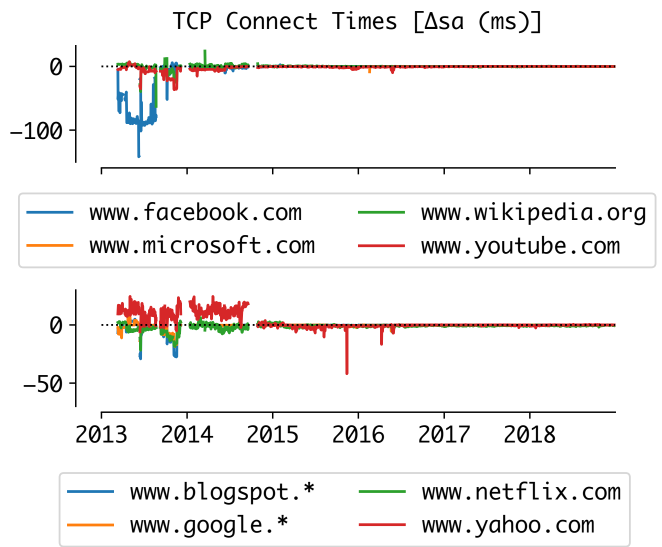

Fig. 18. *Time series of difference in TCP connect times to dual-stacked websites. Latency to popular websites over IPv6 has improved over time.*

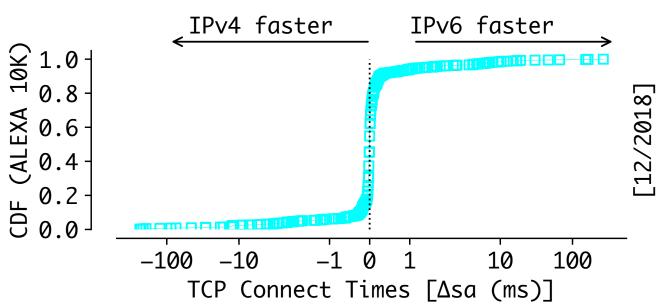

Fig. 19. *CDF of absolute difference of TCP connect times between IPv4 and IPv6 as of December 2018. 56% of the dual-stacked websites with ALEXA rank above 10K are faster over IPv6, 95% of the rest are at most 1 ms slower.*

service provider (undisclosed). Our analysis using more diverse vantage points reveals that `www.facebook.com` connects as fast over IPv6 as over IPv4. Note, SamKnows probes perform measurements only in the absence of cross-traffic [57], [58], as a result the observed latency is not affected by background traffic in the home network.

Summary — A large fraction (∼66%) of ALEXA top 10K websites with AAAA entries are served by CDN deployments over IPv4, with ∼60% of these websites served by CDN deployments over IPv6. Cloudflare (∼35%), Google (∼16%) and Akamai (∼9%) are leading players that contribute to the CDN penetration. The latency towards ALEXA top 10K websites with AAAA entries have reduced in six years by 29% over IPv4 and by 57% over IPv6. As of Dec 2018, ∼56% of dual-stacked websites with ALEXA rank above 10K are faster over IPv6 with ∼95% of the rest at most 1 ms slower and ∼2% of the websites being 25 ms or more slower over IPv6. While, ∼8% of the websites are 1 ms or more faster, with ∼2% being 10 ms or more faster over IPv6.

VII. Happy Eyeballs

These longitudinal observations also help us identify areas of improvements in the IETF standards work. For instance, HE [16] (2017) algorithm when combined with the default address selection policy [22] (2012), gives a noticeable advantage (250 ms, see § II for details) to connections made over IPv6. HE [17] was defined during a time when IPv6 brokenness was quite prevalent, which made applications stall for several seconds before attempting a connection over IPv4. For instance, Savolainen *et al.* [23] (2011) have reported browser connection timeouts to be in the order of 20 seconds. HE timer allowed applications to fast fallback to IPv4 in such situations. The IPv6 brokenness has been largely attributed to failures caused by Teredo [69] and 6to4 relays [70]. Studies [1], [33] (2012, 2010) have shown that even in situations where relays work, Teredo / 6to4 add noticeable latency when compared to native IPv4 and IPv6. With considerable efforts made by the IPv6 operations community, these transition mechanisms appear to steadily decline (see Fig. 6) over the last six years. For instance, Microsoft stopped Teredo on Windows and deactivated [71] its public Teredo servers in 2014. The 6to4 anycast prefix recently has been obsoleted [72] (2015) and future products are recommended to not use 6to4 anycast anymore. Huston [73] (2016) recently showed that as a consequence, failure rates over IPv6 have dropped from 40% (2011) to 3.5% (2015). In fact unicast IPv6 failure rates have also gone down from 5.3% (2011) to 2% (2015). We also observe [74] that HE never prefers IPv6 using Teredo except in situations where IPv4 reachability of the destination endpoint is broken. We investigate the effects of the HE timer value (250 ms) on the overall experience of a dual-stacked user. For instance, it is unclear how often HE makes a deliberate decision of choosing IPv6 when it's slower and in such situations what is the amount of imposition (in terms of latency) a dual-stacked user has to pay as a result of the high HE timer value. This is critical since applications on top of TCP not only apply HE in scenarios where IPv6 connectivity is broken, but also in scenarios where IPv6 connectivity is comparable to IPv4.

In order to address this question, we measured the effects of the HE algorithm. We witnessed that within the last six years, only ∼3% of the samples over IPv6 exhibit TCP connect times above the HE timer (250 ms) value as shown in Fig. 16. In fact 90% of the samples over IPv6 are below 100 ms with 82% of the samples below 50 ms. Similarly, 86% of samples over IPv4 are below 50 ms with 75% below 30 ms. We calculated the preference using the HE timer (250 ms) value over the duration of the dataset. We observed that during the last six years, all probes (sources) preferred IPv6 at least 96% of the time with 99% of probes preferring it more than 98% of the time. Similarly TCP connections over IPv6 to 99% of websites (destinations) were preferred more than 96% of the time. We can conclude that with a HE 250 ms advantage, a dual-stack host tends to use IPv4 connections only ∼4% of the time. Zander *et al.* [32] showed that 20% of the hosts in 2012 had a HE implementation, out of which 75% of the connection attempts preferred IPv6. Fig. 21 shows that this preference (due to decreased latencies over IPv6) has increased to 96%.

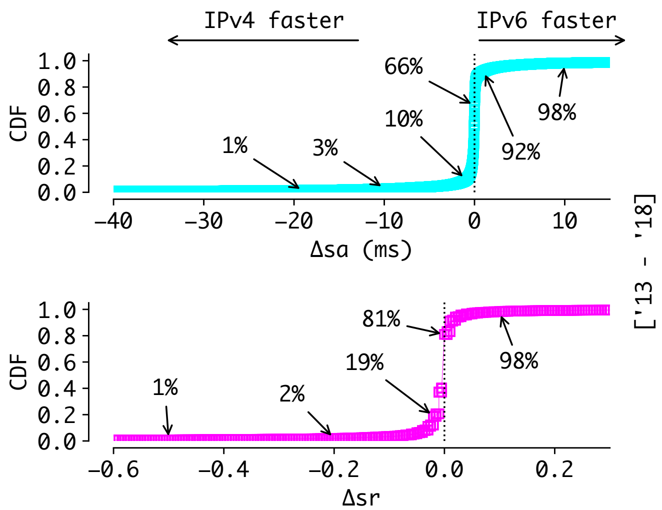

Fig. 20. *CDF of absolute (above) and relative (below) difference of TCP connect times (1M data points) over IPv4 and IPv6 for situations where HE prefers IPv6 using 250 ms timer value. HE prefers slower IPv6 connection in ∼81% cases, but absolute TCP connect times are not far apart from IPv4.*

We further calculated relative and absolute difference in TCP connect times for situations where HE prefers IPv6 using the 250 ms timer value. Fig. 20 shows the relative difference ($\Delta s_r(u) = \frac{\Delta s_a(u)}{t_4(u)}$) for situations where HE prefers IPv6 using the 250 ms timer value. Note, this only includes cases where HE prefers connections over IPv6. The positive values on x-axis represent samples where IPv6 is faster which is ∼19% of the total samples. IPv6 is more than 10% faster in ∼2% of the samples. On the other hand, IPv6 is more than 2% slower in ∼19% of the samples with being more than 20% slower in 2% of the samples. Worse, it is more than 50% slower in 1% of the samples. Fig. 20 also shows the corresponding absolute difference. It can be seen that ∼8% of the samples exhibit TCP connect times that are at least 1 ms faster over IPv6 with ∼2% samples that are at least 10 ms faster. On the other hand, ∼10% of the samples are at least 1 ms slower with 3% of samples that are at least 10 ms slower. In fact only 2% of the samples are at least 12 ms slower with 1% samples being at least 19 ms slower over IPv6. As such, IPv6 may be slower in 81% of the cases where HE prefers it, but the TCP connect times are not that far apart from IPv4. We know that a 250 ms timer value leaves ∼4% chance for IPv4 to win a HE race. In 81% of these cases, HE tends to prefer slower IPv6 connections. We have also seen that HE strongly prefers (more than 97%) connections made over IPv6 for streaming YouTube [51] even though this preference to IPv6 brings worse performance in comparison with IPv4. The high HE timer value is the reason why we observe fragmentation of the algorithm (see § II) in browser implementations.

We experimented by lowering the HE timer advantage. We know that by using 250 ms HE timer, IPv6 connections to 99% of ALEXA websites are preferred more than 96% of the time. The idea towards finding a better HE timer value

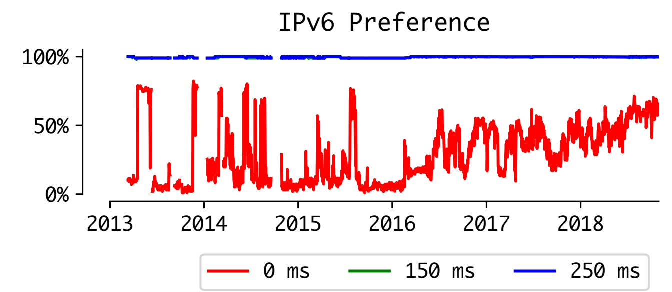

Fig. 21. *Time series (gaps represent missing data) of IPv6 preference to dual-stacked websites. Disabling HE entirely (0 ms) hampers IPv6 preference.*

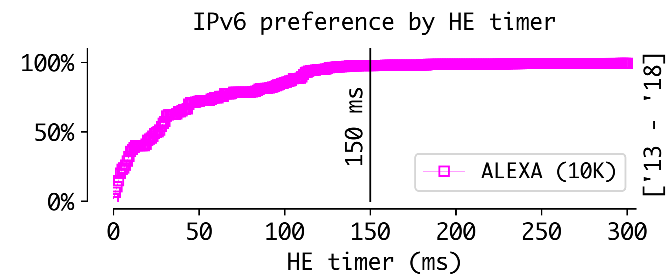

Fig. 22. *TCP connection establishment preference over IPv6 towards websites by varying the HE timer value. A HE timer value of 150 ms allows IPv6 connections to 99% of websites to be preferred at least 96% of the time.*

is to control these two parameters (IPv6 connections to 99% websites are preferred 96% of the time) and lower the HE timer value to see until when this precedence remains true. This is important because the timer value cannot be lowered to zero (parallel connections over IPv4 and IPv6), since HE must still adhere to the IPv6 upgrade policy (see § II) to prefer IPv6 paths. Fig. 21 shows that disabling HE entirely by using parallel TCP connections (such as used by Mozilla Firefox and Opera) hampers preference to IPv6 since only ∼56% of dual-stacked websites with ALEXA rank above 10K are faster (see Fig. 19) over IPv6 as of Dec 2018. As such, the timer value by design should give IPv6 a fair chance to succeed, but at the same time reduce wait time for a dual-stack host in situations where IPv6 is considerably slower. Fig. 22 shows TCP connection establishment preference over IPv6 by varying the HE timer value. Each data point represents how often (over the six-year long duration of the dataset) IPv6 connections from 102 probes towards 99% of ALEXA 10K websites are preferred at least 96% of the time. As can be seen, a HE timer value of 150 ms allows IPv6 connections to same 99% of the websites to be still preferred at least 96% of the time. Note, the sample of probes after splitting observations by region or by network also goes down significantly. As such, it becomes difficult to further reasonably discuss latency distributions (or HE timer implications) by region or by network.

Summary — The reduced brokenness over IPv6 due to decline of Teredo and 6to4 transitioning mechanisms over the years and comparable performance of IPv6 creates an *opportunity to lower the HE timer value. Our measurements show that we are able to reduce the HE timer value by half retaining similar preference to connections over IPv6.*

VIII. Lessons Learned

We reflect on some of the lessons learned from this study:

1. Sustaining the longitudinal measurements: Volunteers had apprehensions that the probe can be used to passively snoop on their home traffic. We found that leveraging current users to spread awareness helps create a web of trust to get more volunteers on board. Active engagement with the users also helps sustain the deployment long term. Users are happy to keep the probe connected once they see the value. In certain situations, we also helped users by providing insights from our data that they could relay back to their ISP. The results presented in this paper highlight the importance of data collection over a longitudinal period to understand the evolution of a protocol adoption on the Internet.

2. Collecting additional metrics for data interpretation: Results reported by a metric require cross-correlation with results from additional metrics to provide a good view on the state of the network. For instance, in addition to TCP connect times, reverse DNS entries and forwarding paths provide essential information to analyze observed latency differences. We now provision additional tests to capture this information, but in hindsight it would have been nice to have this data collected since the beginning of our measurement activity.

3. Metrics that provide a more holistic view of failures: We observed popular dual-stacked websites exhibiting partial failures over IPv6 due to missing `AAAA` entries for content embedded within webpages. Metrics that limit measurements to the root webpage fail to identify such partial failures over IPv6. These failures silently exist since users also do not notice them as long as the content can still be accessed over IPv4. As such, monitoring platforms should develop and adopt metrics that can measure the entire content of a website to gather a more holistic perspective on the state of IPv6 adoption.

4. Impact of large CDN players on IPv6 adoption: A large fraction of dual-stacked websites are hosted by large CDNs. Akamai has shown that amongst top 25 customer networks by traffic volume, 14 networks have over 10% IPv6 adoption as of June 2016. As such, a CDN can play a leading role in not only pushing technology adoption, but also shifting significant traffic overnight towards IPv6. Cloudflare has also shown that nearly 25% of their IPv6 traffic gets delivered to mobile devices as of June 2016. This shows that mobile networks are starting to play a lead role in this evolution too. In the future, we plan to also extend our measurements to cover mobile networks.

IX. Conclusion

We witness that one quarter of the connections made to Google get established over IPv6 as of Jan 2019. More of these connections are established from residential networks, indicating an increase in IPv6 deployment towards the edge of the network, with a decrease in the adoption of transitioning technologies and deployment of native IPv6 by ISPs.

Meanwhile, web content delivery is largely driven by CDN providers, whereby we witness that more than half of ALEXA top 10K websites with `AAAA` entries are served by CDN deployments led by major players such as Cloudflare, Google and Akamai as of 2018. This increased CDN penetration has brought down failures in content delivery over IPv6 to around 2% and reduced latency over IPv6 by more than half in the past six years. As of 2018, more than half of the ALEXA top 10K websites with `AAAA` entries are faster over IPv6 while only 2% are 25 ms or more slower than IPv4. We believe that our results help shed light to the notion that IPv6 can longer be deemed a second-class citizen on the Internet.

Our longitudinal measurements also facilitate towards concretising the HE timer value by showing that HE with a 150 ms timer value does not severly affect IPv6 preference towards websites. In fact, HE with a 150 ms timer value could have served since the beginning when RFC 6555 was defined. As such, measurements should be actively used by the standards community to inform protocol engineering and practice.

We observed that a few websites which used to be IPv6 capable once did not remain as such forever. We witnessed cases where a dual-stacked website stopped announcing `AAAA` entries in DNS over time. We also showed that metrics that limit only to the root webpage of a dual-stacked website can lead to an overestimation of IPv6 adoption numbers on the Internet. We witnessed several cases where images, javascript and CSS content of a dual-stacked website did not have `AAAA` entries in the DNS. We recommend a stricter policy where a website is deemed IPv6 ready only when there is no partial failure to fetch its content over IPv6. The Internet Society is now supporting the development of tools (`nat64check.ipv6-lab.net`) that help identify such partial failures and the IETF IP Performance Metrics (IPPM) working group is expanding the coverage of its active metrics to include IPv6.

Future Work: The subjective analysis of witnessed partial failures over IPv6 and their effects on the user experience requires further investigation. The clients can also be made to adaptively change the HE timer value based on the previously witnessed history of the TCP connect times as described in RFC 8305 [16] as one of the augmentations to the algorithm. It remains to be seen whether browser implementations prefer to trade complexity for an increased intelligence in the future.

Reproducibility Considerations: The `happy` and `simweb` test and Jupyter notebooks used in the analysis are open-sourced. The entire dataset is publicly released [21]. Guidance on how to reproduce [75], [76] results will be provided and reproducers are encouraged to contact authors for questions.

X. Acknowledgements

We thank SamKnows volunteers, Sam Crawford, Jamie Mason, Steffie Jacob Eravuchira, Jörg Ott, Dan Wing and Andrew Yourtchenko for contributing to this study. The results presented on complete failures are based on a publicly available dataset collected by Dan Wing.

REFERENCES

[1] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice, "Evaluating IPv6 Adoption in the Internet," ser. PAM '10, 2010, pp. 141–150. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12334-4_15

[2] A. Dhamdhere, M. J. Luckie, B. Huffaker, kc claffy, A. Elmokashfi, and E. Aben, "Measuring the Deployment of IPv6: Topology, Routing and Performance," ser. IMC '12, 2012. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398832

[3] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, "Measuring IPv6 Adoption," ser. ACM SIGCOMM, 2014. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626295

[4] M. Nikkhah, R. Guérin, Y. Lee, and R. Woundy, "Assessing IPv6 Through Web Access A Measurement Study and Its Findings," ser. ACM CoNEXT '11, 2011, p. 26. [Online]. Available: http://doi.acm.org/10.1145/2079296.2079322

[5] "Google IPv6 Adoption Statistics," http://www.google.com/intl/en/ipv6/statistics.html, [Online; accessed 25-Mar-2017].

[6] "Internet Society - World IPv6 Day," http://worldipv6day.org, [Online; accessed 25-Mar-2017].

[7] "Internet Society - World IPv6 Launch," http://www.worldipv6launch.org, [Online; accessed 25-Mar-2017].

[8] P. Richter, M. Allman, R. Bush, and V. Paxson, "A Primer on IPv4 Scarcity," ser. ACM CCR, vol. 45, no. 2, 2015, pp. 21–31. [Online]. Available: http://doi.acm.org/10.1145/2766330.2766335

[9] "World IPv6 Launch - Measurements," http://www.worldipv6launch.org/measurements, [Online; accessed 25-March-2017].

[10] "Comcast Reaches Key Milestone in Launch of IPv6 Broadband Network," https://goo.gl/thZo4U, [Online; accessed 25-Mar-2017].

[11] "NANOG - IPv6 traffic percentages?" http://goo.gl/w1mJ8y, [Online; accessed 20-Sep-2017].

[12] I. Livadariu, A. Elmokashfi, and A. Dhamdhere, "Characterizing IPv6 control and data plane stability," ser. IEEE INFOCOM, 2016. [Online]. Available: http://dx.doi.org/10.1109/INFOCOM.2016.7524465

[13] U. Goel, M. Steiner, M. P. Wittie, E. Nygren, M. Flack, and S. Ludin, "A Case for Faster Mobile Web in Cellular IPv6 Networks," ser. MobiCom, 2016. [Online]. Available: http://doi.acm.org/10.1145/2973750.2973771

[14] E. Pujol, P. Richter, and A. Feldmann, "Understanding the Share of IPv6 Traffic in a Dual-Stack ISP," ser. PAM, 2017, pp. 3–16. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-54328-4_1

[15] Geoff Huston, "IPv6 Performance - Revisited," http://www.potaroo.net/ispcol/2016-08/v6perf.html, [Online; accessed 25-Mar-2017].

[16] D. Schinazi and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency," RFC 8305, IETF, 2017. [Online]. Available: https://doi.org/10.17487/RFC8305

[17] D. Wing and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts," RFC 6555, IETF, 2012, https://tools.ietf.org/html/rfc6555.

[18] V. Bajpai and J. Schönwälder, "IPv4 versus IPv6 - who connects faster?" ser. IFIP NETWORKING '15, 2015, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/IFIPNetworking.2015.7145323

[19] ——, "Measuring the Effects of Happy Eyeballs," ser. Applied Networking Research Workshop (ANRW), 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2959429

[20] S. J. Eravuchira, V. Bajpai, J. Schönwälder, and S. Crawford, "Measuring Web Similarity from Dual-stacked Hosts," ser. CNSM, 2016. [Online]. Available: http://dx.doi.org/10.1109/CNSM.2016.7818415

[21] V. Bajpai and J. Schönwälder. (2019) A Longitudinal View of Dual-stacked Websites - Failures, Latency and Happy Eyeballs (Software and Dataset). [Online]. Available: https://github.com/vbajpai/2019-ton-v6-websites-analysis

[22] D. Thaler, R. Draves, A. Matsumoto, and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)," RFC 6724, Internet Engineering Task Force, Sep. 2012, https://tools.ietf.org/html/rfc6724.

[23] "Teemu Savolainen - Experiences of host behavior in broken IPv6 networks," http://www.ietf.org/proceedings/80/slides/v6ops-12.pdf, v6ops WG, IETF 80, [Online; accessed 25-Mar-2017].

[24] "StatCounter - Desktop Browser Market Share," https://goo.gl/gmhb9D, [Online; accessed 21-Jan-2019].

[25] "W3Counter - Desktop Browser Market Share," https://www.w3counter.com/globalstats.php, [Online; accessed 21-Jan-2019].

[26] "NetMarketShare - Desktop Browser Market Share," https://goo.gl/uzwufA, [Online; accessed 21-Jan-2019].

[27] "Google Chrome - Revision 85934: Add a fallback socket connect() for IPv6." http://src.chromium.org/viewvc/chrome?view=rev&revision=85934, [Online; accessed 25-Mar-2017].

[28] "Mozilla Firefox 15 - Release Notes," http://www.mozilla.org/en-US/firefox/15.0/releasenotes/buglist.html, [Online; accessed 25-Mar-2017].

[29] "Opera 12.10 - Changelog," http://www.opera.com/docs/changelogs/unified/1210, [Online; accessed 25-Mar-2017].

[30] "Apple - Lion and IPv6," http://lists.apple.com/archives/ipv6-dev/2011/Jul/msg00009.html, [Online; accessed 25-Mar-2017].

[31] "Apple and IPv6 - Happy Eyeballs," https://www.ietf.org/mail-archive/web/v6ops/current/msg22455.html, [Online; accessed 25-Mar-2017].

[32] S. Zander, L. L. H. Andrew, G. J. Armitage, G. Huston, and G. Michaelson, "Mitigating Sampling Error when Measuring Internet Client IPv6 Capabilities," ser. Internet Measurement Conference (IMC), 2012. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398787

[33] ——, "Investigating the IPv6 Teredo Tunnelling Capability and Performance of Internet Clients," ser. ACM CCR '12, 2012. [Online]. Available: http://doi.acm.org/10.1145/2378956.2378959

[34] M. Karir, G. Huston, G. Michaelson, and M. Bailey, "Understanding IPv6 Populations in the Wild," ser. PAM '13, 2013, pp. 256–259. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36516-4_27

[35] A. W. Berger, N. Weaver, R. Beverly, and L. Campbell, "Internet nameserver IPv4 and IPv6 address relationships," ser. IMC, 2013. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504745

[36] R. Beverly and A. W. Berger, "Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure Via Active Fingerprinting," ser. PAM, 2015. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15509-8_12

[37] V. Giotsas, M. J. Luckie, B. Huffaker, and kc claffy, "IPv6 AS Relationships, Cliques, and Congruence," ser. PAM, 2015, pp. 111–122. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15509-8_9

[38] M. Nikkhah and R. Guérin, "Migrating the Internet to IPv6: An Exploration of the When and Why," ser. IEEE/ACM TON, 2016. [Online]. Available: http://dx.doi.org/10.1109/TNET.2015.2453338

[39] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczynski, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," ser. IMC, 2018, pp. 364–378. [Online]. Available: http://doi.acm.org/10.1145/3278532.3278564

[40] K. Fukuda and J. S. Heidemann, "Who Knocks at the IPv6 Door?: Detecting IPv6 Scanning," ser. IMC, 2018, pp. 231–237. [Online]. Available: http://doi.acm.org/10.1145/3278532.3278553

[41] J. P. Rohrer, B. LaFever, and R. Beverly, "Empirical Study of Router IPv6 Interface Address Distributions," ser. IEEE Internet Computing, 2016. [Online]. Available: http://dx.doi.org/10.1109/MIC.2016.52

[42] O. Gasser, Q. Scheitle, S. Gebhard, and G. Carle, "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist," ser. IFIP TMA, 2016. [Online]. Available: http://tma.ifip.org/2016/papers/tma2016-final51.pdf

[43] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna, "Something from Nothing (There): Collecting Global IPv6 Datasets from DNS," ser. PAM, 2017, pp. 30–43. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-54328-4_3

[44] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, "In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery," ser. IMC, 2018, pp. 308–321. [Online]. Available: http://doi.acm.org/10.1145/3278532.3278559

[45] D. Plonka and A. W. Berger, "Temporal and Spatial Classification of Active IPv6 Addresses," ser. ACM IMC, 2015, pp. 509–522. [Online]. Available: http://doi.acm.org/10.1145/2815675.2815678

[46] P. Foremski, D. Plonka, and A. W. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses," ser. ACM IMC, 2016, pp. 167–181. [Online]. Available: http://dl.acm.org/citation.cfm?id=2987445

[47] J. Czyz, M. J. Luckie, M. Allman, and M. Bailey, "Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy," ser. NDSS, 2016. [Online]. Available: http://www.internetsociety.org/events/ndss-symposium-2016

[48] L. Hendriks, R. Schmidt, R. van Rijswijk-Deij, and A. Pras, "On the Potential of IPv6 Open Resolvers for DDoS Attacks," ser. PAM, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-54328-4_2

[49] H. A. Alzoubi, M. Rabinovich, and O. Spatscheck, "Performance Implications of Unilateral Enabling of IPv6," ser. PAM, 2013, pp. 115–124. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36516-4_12

[50] S. Ahsan, V. Bajpai, J. Ott, and J. Schönwälder, "Measuring YouTube from Dual-Stacked Hosts," ser. PAM, 2015, pp. 249–261. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15509-8_19

[51] V. Bajpai, S. Ahsan, J. Schönwälder, and J. Ott, "Measuring YouTube Content Delivery over IPv6," ser. Computer Communication Review, vol. 47, no. 5, 2017, pp. 2–11. [Online]. Available: http://doi.acm.org/10.1145/3155055.3155057

[52] T. V. Doan, L. Pajević, V. Bajpai, and J. Ott, "Tracing the Path to YouTube: A Quantification of Path Lengths and Latencies toward Content Caches," ser. IEEE Communications Magazine, 2019. [Online]. Available: https://doi.org/10.1109/MCOM.2018.1800132

[53] "Emile Aben - Hampering Eyeballs: Observations on Two Happy Eyeballs Implementations," RIPE Labs: https://labs.ripe.net/Members/emileaben/hampered-eyeballs, [Online; accessed 25-Mar-2017].

[54] "Geoff Huston - Dual Stack Esotropia," APNIC Labs: http://labs.apnic.net/blabs/?p=47, [Online; accessed 25-Mar-2017].

[55] "Geoff Huston - Bemused Eyeballs: Tailoring Dual Stack Applications for a CGN Environment," The ISP Column: http://www.potaroo.net/ispcol/2012-05/notquite.html, [Online; accessed 25-Mar-2017].

[56] F. Baker, "Testing Eyeball Happiness," RFC 6556 (Informational), IETF, Apr. 2012. [Online]. Available: https://tools.ietf.org/html/rfc6556

[57] V. Bajpai, S. J. Eravuchira, and J. Schönwälder, "Lessons Learned From Using the RIPE Atlas Platform for Measurement Research," ser. Computer Communication Review (CCR), vol. 45, no. 3, 2015, pp. 35–42. [Online]. Available: http://doi.acm.org/10.1145/2805789.2805796

[58] V. Bajpai and J. Schönwälder, "A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts," ser. IEEE Communications Surveys and Tutorials, 2015. [Online]. Available: http://dx.doi.org/10.1109/COMST.2015.2418435

[59] V. Bajpai, S. J. Eravuchira, J. Schönwälder, R. Kisteleki, and E. Aben, "Vantage point selection for IPv6 measurements: Benefits and limitations of RIPE Atlas tags," ser. IFIP/IEEE Symposium on Integrated Network and Service Management, 2017, pp. 37–44. [Online]. Available: https://doi.org/10.23919/INM.2017.7987262

[60] B. N. Chun, D. E. Culler, T. Roscoe, A. C. Bavier, L. L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," ser. CCR '03, vol. 33, no. 3, 2003, pp. 3–12. [Online]. Available: http://doi.acm.org/10.1145/956993.956995

[61] S. D. Strowes, "Diurnal and Weekly Cycles in IPv6 Traffic," ser. Applied Networking Research Workshop, ANRW, 2016, pp. 65–67. [Online]. Available: http://dl.acm.org/citation.cfm?id=2959438

[62] "[v6ops] On Google IPv6 statistics," https://goo.gl/r9ybxn, [Online; accessed 26-June-2018].

[63] "Dan Wing - AAAA and IPv6 Connectivity Statistics," http://www.employees.org/~dwing/aaaa-stats, [Online; accessed 25-Mar-2017].

[64] "98.01% of sites on Cloudflare now use IPv6," https://blog.cloudflare.com/98-percent-ipv6, [Online; accessed 15-Apr-2017].

[65] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing," RFC 7230, Internet Engineering Task Force, Jun. 2014. [Online]. Available: https://tools.ietf.org/html/rfc7230

[66] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Characterizing Web Page Complexity and Its Impact," ser. IEEE/ACM TON, 2014. [Online]. Available: http://dx.doi.org/10.1109/TNET.2013.2269999

[67] ——, "Understanding Website Complexity: Measurements, Metrics, and Implications," ser. Internet Measurement Conference (IMC), 2011. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068846

[68] D. Gugelmann, B. Ager, and V. Lenders, "Towards Classifying Third-Party Web Services at Scale," ser. CoNEXT Student Workshop '14, 2014. [Online]. Available: http://doi.acm.org/10.1145/2680821.2680838

[69] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network NATs," RFC 4380, IETF, Feb. 2006, https://tools.ietf.org/html/rfc4380.

[70] B. Carpenter and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," RFC 3056, IETF, Feb. 2001, https://tools.ietf.org/html/rfc3056.

[71] "Christopher Palmer - Teredo at Microsoft: Present and Future," http://www.ietf.org/proceedings/88/slides/slides-88-v6ops-0, v6ops WG, IETF 88, [Online; accessed 25-Mar-2017].

[72] O. Troan and B. Carpenter, "Deprecating the Anycast Prefix for 6to4 Relay Routers," RFC 7526, Internet Engineering Task Force, May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7526

[73] "Geoff Huston - Measuring IPv6 Performance," https://goo.gl/n78W1t, NANOG 66, [Online; accessed 10-February-2016].

[74] V. Bajpai and J. Schönwälder, "Measuring the Effects of Happy Eyeballs," Internet Engineering Task Force, Internet-Draft, Jul. 2013. [Online]. Available: http://tools.ietf.org/html/draft-bajpai-happy-01

[75] V. Bajpai, A. Brunstrom, A. Feldmann, W. Kellerer, A. Pras, H. Schulzrinne, G. Smaragdakis, M. Wählisch, and K. Wehrle, "The Dagstuhl Beginners Guide to Reproducibility for Experimental Networking Research," ser. ACM CCR, 2019 (to appear).

[76] V. Bajpai, M. Kühlewind, J. Ott, J. Schönwälder, A. Sperotto, and B. Trammell, "Challenges with Reproducibility," ser. SIGCOMM Reproducibility Workshop, 2017, pp. 1–4. [Online]. Available: http://doi.acm.org/10.1145/3097766.3097767

Vaibhav Bajpai is postdoctoral fellow at TUM, Germany. He received his PhD (2016) and Masters (2012) degrees in Computer Science from Jacobs University Bremen, Germany. He is the recipient of the ACM SIGCOMM (2018) best paper award and IEEE COMSOC award (2017) for the best dissertation in network and service management. He is interested in future Internet protocols, content delivery, network management and reproducibility of scientific research.

Jürgen Schönwälder is professor of computer science at Jacobs University Bremen. His research interests include network management and measurement, network security, embedded systems, and distributed systems. He is an active member of the IETF where he has co-authored more than 40 specifications and standards. He has been PI in several European research projects. Jürgen Schönwälder serves as co-editor of the Network and Service Management series of the IEEE Communications Magazine.