A Longitudinal View of Netflix: Content Delivery over IPv6 and Content Cache Deployments

Trinh Viet Doan
Technical University of Munich
doan@in.tum.de

Vaibhav Bajpai
Technical University of Munich
bajpaiv@in.tum.de

Sam Crawford
SamKnows
sam@samknows.com

***Abstract*—We present an active measurement test (`netflix`) that downloads content from the Netflix content delivery network. The test measures latency and achievable throughput as key performance indicators when downloading the content from Netflix. We deployed the test on ∼100 SamKnows probes connected to dual-stacked networks representing 74 different origin ASes. Using a ∼2.75 year-long (Jul 2016–Apr 2019) dataset, we observe Netflix Open Connect Appliance (OCA) infrastructure to be highly available, although some vantage points experience low success rates connecting over IPv6. We witness that clients prefer connecting to Netflix OCAs over IPv6, although the preference over IPv6 tends to drop over certain peak hours during the day. The TCP connect times toward the OCAs have reduced by ∼40% and the achievable throughput has increased by 20% over the measurement duration. We also provision `scamper` right after the `netflix` test to capture the forwarding path toward the Netflix OCAs. We observe that the Netflix OCA caches deployed inside the ISP are reachable within six IP hops and can reduce IP path lengths by 40% over IPv4 and by half over IPv6. Consequently, TCP connect times are reduced by ∼64% over both address families. The achieved throughput can also increase by a factor of three when such ISP caches are used to stream content. This is the first study to measure Netflix content delivery from residential networks, since the inception of the Netflix CDN infrastructure in 2011. To encourage reproducibility of our work, an anonymized version of the entire longitudinal dataset is publicly released.**

I. Introduction

Netflix and Youtube are the leading content delivery services that contribute toward more than half of the downstream traffic, with Netflix, being the single largest [1], [2] source of downstream traffic (∼35%) in North America. This is attributed to increasing demand of high fidelity content (4K and HDR video streaming) and emerging applications (augmented and virtual reality) that further tend to push an increase of this traffic composition. In order to meet growing demands of a large consumer base of more than 100M subscribers [3], Netflix (starting 2011) launched and now operates Open Connect, a Content Delivery Network (CDN) infrastructure that consists of Open Connect Appliances (OCAs) deployed close to the users. Netflix uses Amazon Web Service (AWS) as a control plane to route requests to the nearest OCA based on the network conditions, device type, and location of the user. Given that OCAs are also dual-stacked [3], the content can be delivered over IPv6 in situations where the ISP provides IPv6 connectivity to its users. Netflix reports that ∼13% of the session hours [3] measured globally are over IPv6. As such, we want to know – *How does Netflix content delivery perform (regarding latency and achieved throughput) over both address families? Particularly, do users get benefits (or suffer) from downloading Netflix content over IPv6 compared to IPv4?*

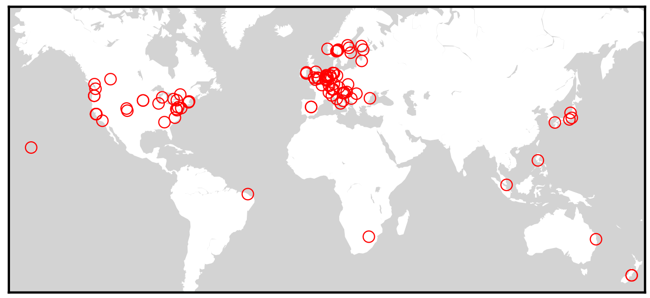

Fig. 1. Geographical distribution of ∼100 dual-stacked SamKnows probes, with ∼80% of these probes being deployed at home. The metadata for each probe is made publicly available together with the measurement data.

There are multiple strategies (§II) employed to stream content from Netflix OCA deployments. ISPs can stream content from the OCAs either by privately peering with Netflix via one (or more) transit providers, or ISPs can peer directly at public Internet Exchange Points (IXPs) [4], [5], [6]. ISPs can also embed an OCA directly inside the ISP network to circumvent the possibility of video degradation due to persistent congestion [7] on interdomain links. As such, in situations where the OCA is deployed within the ISP boundary (which we refer to as content caches and ISP caches hereafter), we also want to know: *To what extent do content caches deployed at the edge benefit Netflix content delivery? How do these benefits compare over IPv4 and IPv6? How far are these caches in terms of IP path lengths and latency when compared to OCA deployments outside the ISP boundary? How do path lengths and latency differ over IPv4 and IPv6?*

Toward these ends, we developed `netflix`, an active test that measures Netflix content delivery over both address families. We deployed this test on ∼100 geographically distributed SamKnows [8] probes (Fig. 1) to provide diversity of network origins. These probes receive native dual-stacked connectivity and belong to different ISPs covering 74 different origin ASes. The key **contributions** of this paper include the `netflix` test and findings observed by analyzing the longitudinal dataset as summarized below –

netflix, an active measurement test written in C. The test downloads content from the Netflix content delivery network (§III) and measures TCP connect times and achievable throughput as key performance indicators when downloading content from Netflix.

We also provision `scamper` [9] immediately after the `netflix` test completes to capture the forwarding path toward the OCA destinations identified by the `netflix` test. We perform analysis using a ∼2.75 year-long (July 2016–Apr 2019) dataset collected using these metrics from the dual-stacked probes. To encourage reproducibility [10] (§VII) of our work, an anonymized version of the entire longitudinal dataset is publicly released. Our findings are –

Content Delivery over IPv6 (§IV) – We witness that besides some vantage points that experience low success rates when connecting to Netflix OCAs over IPv6, the OCA infrastructure appears to be highly available. We observe that clients prefer connecting to the Netflix OCAs over IPv6, although the preference tends to drop over peak hours in the day. The latency toward the Netflix OCAs has reduced over the years with TCP connect times being comparable over both address families and dropping by roughly 10 ms (−40%) over the measurement duration. Achievable throughput has increased over the year, initially being ∼10 MB/s and increasing to ∼12 MB/s (+20%) over both address families. For around 56% of the samples, IP paths toward Netflix OCAs were even, while 21.5% were shorter over IPv4 (22.5% shorter over IPv6).

Content Cache Deployments (§V) – We observe that Netflix OCAs deployed inside the ISP can reduce TCP connect times by up to ∼64% over both address families. In 90% of the cases, throughput toward an ISP cache was higher over both address families in comparison with a CDN OCA. When quantifying the benefit of the caches, the achieved throughput can be increased by a factor of up to three for 75% of the samples, from roughly 11 MB/s to 33 MB/s. These content caches are reachable within six IP hops and can reduce IP path lengths by 40% over IPv4 and by half over IPv6.

II. Background and Related Work

Fig. 2 presents a simplified architecture of the Netflix video streaming platform and experiment setup. It consists of three key components: clients that request content (SamKnows probes within the context of this paper), the Netflix control plane hosted in the AWS cloud, and a swarm of Netflix OCA servers that constitute the Open Connect CDN infrastructure. The clients authenticate with the application (Netflix's `fast.com` within the context of this paper) hosted in the AWS. On successful authentication, the application returns a manifest containing a list of URLs of different OCAs that can serve the content. This behavior of `fast.com` matches the steering logic used for `netflix.com`, as both will refer to the same Netflix OCA for a request. The returned OCAs are chosen based on the proximity and traffic load on each OCA. The client finally requests content from one of the OCAs provided in the manifest.

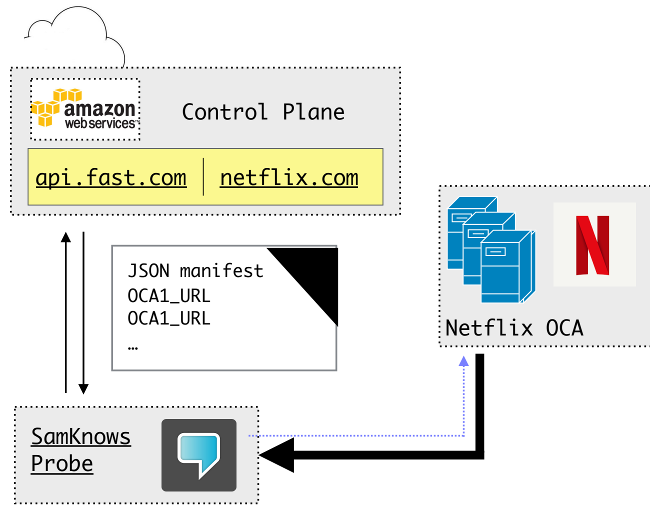

Fig. 2. A simplified architecture of Netflix content delivery. Netflix leverages AWS as a control plane to offer a list of the most appropriate OCA servers to serve the request and deliver the content.

Related Work – Adhikari *et al.* [11] study the Netflix architecture and its service strategy. They show that Netflix uses AWS as a control plane to route requests to third-party CDNs for delivering multimedia content. They propose to improve video content delivery using multiple CDNs, by choosing the best performing CDN based on measurements conducted in the beginning of the video playback. They show that this improves the bandwidth by ∼12% over a single CDN, with more than 50% bandwidth improvement in case of leveraging multiple CDNs. Adhikari *et al.* [12] take this further and compare Netflix and Hulu CDNs, revealing that both platforms heavily depend on third-party infrastructure (AWS or Akamai) for content delivery. It was observed that the steering logic only takes the origin of the request into account when selecting the most suitable CDN replica for serving the request. Martin *et al.* [13] characterize the bandwidth consumption of Netflix as an example of a widely deployed Dynamic Adaptive Streaming over HTTP (DASH) application. They observed that during periods of heavy, sustained network congestion, Netflix adaptation defaults to underlying TCP mechanisms, whereas in situations of unstable network conditions, the DASH adaptation logic underestimates available bandwidth over TCP. Netflix used to leverage third-party CDNs for content delivery in the past, however, these studies have become outdated with time and the deployment of Netflix's Open Connect CDN in 2012.

Huang *et al.* [14] study the rate selection of popular video streaming services (such as Netflix). They show that in the presence of competing TCP flows, Adaptive Bitrate Streaming (ABR) selection algorithms can drop the bitrate lower than what the network can sustain (downward spiral effect). They reason that this is because throughput observed over HTTP can create a bad estimate of available bandwidth when TCP congestion control dynamics are in play with TCP competing flows. They recommend that ABR selection algorithms (using HTTP) should not attempt to estimate bandwidth at all and take this further [15], showing that capacity estimation can be avoided in steady state, and only used during the startup phase when the empty buffer is growing. With respect to Open Connect, Böttger *et al.* [4], [16] recently studied Netflix OCA infrastructure deployment and show that Netflix OCAs are present in nine of the top ten largest IXPs of the world. ISPs can stream content from these OCA deployments by peering directly at public IXPs [17], [18]. Meanwhile, ISPs can also privately peer with Netflix via one (or more) transit providers or embed the OCAs directly inside their own network, similar to Google Global Caches used for YouTube [19]. As can be seen, there has been no study on measuring Netflix content delivery over IPv6, with a focus on observed network characteristics of content cache deployments, particularly since the advent of the Netflix CDN infrastructure in 2012.

III. Methodology

A. Metrics and Implementation

`netflix` – We have developed a test (`netflix`) that downloads content from the Netflix CDN. The test begins by calling a Netflix hosted web-based API, which later evolved into Netflix's `fast.com` API. This API [20] examines the client's source IP endpoint and uses the existing proprietary internal Netflix logic to determine which Netflix server this user's IP endpoint would normally be served content from. This logic takes into account the ISP and geographic location of the requesting IP endpoint. In situations where the ISP participates in Netflix's Open Connect program, the request will get served by an OCA that functions as an edge device, serving content close to the users. The API returns a HTTP 302 redirect to a 25 MB binary file, hosted on the applicable content server, to the client. The test will then establish an HTTP session to the redirected server and attempt to fetch the 25 MB binary file. This runs for a fixed 20 seconds of realtime. HTTP pipelining is used to request multiple copies of the 25 MB binary, ensuring that if the payload is exhausted before the 20 seconds have elapsed, the test can continue receiving more data. The client downloads data at full rate throughout the duration of the test, with no throttling initiated at the client-side at any point. Note that the 25 MB binary content does not contain video or audio but random binary data instead; this is because the `fast.com` API is made available by Netflix to provide speed test functionality only. In this way, the test measures latency and achievable throughput as key performance indicators when downloading content over both address families from the Netflix content delivery network.

`scamper` – We also run `scamper` [9] immediately after the completion of the `netflix` test. The `scamper` test performs `paris-traceroute` [21] over ICMP toward the Netflix content server identified by the `netflix` test (or OCA nodes in situations where the ISP participates in the Open Connect program) both over IPv4 and IPv6.

B. Measurement Setup and Dataset

We deployed the `netflix` and `scamper` tests on $\sim$100 SamKnows probes (Fig. 1) connected in dual-stacked networks representing 74 different origin ASes. Around 80% of these probes are connected in residential networks served by the RIPE and ARIN regional registries. SamKnows [8] 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 Federal Communications Commission (FCC) as part of the Measuring Broadband America (MBA) [22], [23], [24], [25] project. We currently use a pilot deployment of these probes, as the `netflix` test is not deployed on the FCC panel of SamKnows probes yet. The tests run twice, once for IPv4 and subsequently for IPv6, and repeat every hour. The datasets encompass measurements from Jul 2016–Apr 2019, accounting for 11 GB worth of data. The `netflix` dataset consists of 2.1M distinct and successful measurements, with 1.17M being collected over IPv4, and around 970k over IPv6. Due to the design of the test setup, the `scamper` dataset covers 1.1M traced paths over IPv4 and 900k over IPv6, respectively, leading to measurements conducted for overall 2M IP paths. Differences in the number of measurements are seen due to different success rates over the address families.

IV. Content Delivery over IPv6

We begin by comparing the Netflix content delivery over IPv4 and IPv6. We particularly focus on IPv6, as prominent ISPs have rolled out IPv6 in both fixed-line and cellular networks lately due to the rapid depletion of the IPv4 address space [26]. Furthermore, given that dual-stacked clients prefer requesting content over IPv6 [27], [28], [29] rather than IPv4, the amount of IPv6 traffic on the Internet is also increasing with these rollouts. ISPs such as Comcast and Swisscom estimate IPv6 traffic within their network to be a quarter of the total traffic [30]. In terms of traffic volume, this amounts to more than 1 Tbps of native IPv6 traffic, as observed by Comcast. Consequently, content providers witness an increasing trend of clients requesting content over IPv6. For instance, Google reports that 25–30% of the connections made to Google get established over IPv6 as of 2019 [31]. Furthermore, Google reports that $\sim$90% of traffic originating from Verizon Wireless clients is over IPv6 [30]. Similarly, according to Facebook and Akamai, more than half of the traffic from four major US mobile networks to Facebook and dual-stacked websites hosted on Akamai originates from IPv6 clients alone [30]. In particular, Netflix is a leading player contributing to the largest source of IPv6 traffic in ISP networks. Previous studies measuring IPv6 performance largely measured IPv6 content delivery toward dual-stacked websites [32], finding that IPv6 performance is comparable to IPv4, with the disparity being due to routing in the control plane. However, we are not aware of previous work on measuring Netflix content delivery over IPv6. Therefore, we compare the Netflix content delivery over IPv4 and IPv6, using multiple metrics, to investigate this.

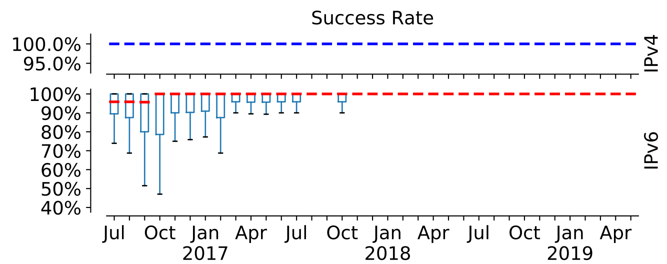

Fig. 3. Boxplot of daily success rates per probe over IPv4 and IPv6, aggregated by month. The Netflix OCA infrastructure is highly available over both address families, though IPv6 shows a higher variance regarding success rates in the first half of the measurement period.

A. Success Rate

We start by comparing the success rate of `netflix` tests over both address families. The test is deemed successful when it successfully downloads the content from Netflix. Note that while the `netflix` test reports stall events (which we do not analyze in this study but identify for future work) as errors, we discard stall-related errors in our analysis. Overall, we observe ∼11.7% failed measurements (282k). The majority of the failures (irrespective of the address family) occurred on the client-side, i.e., closer to the probes. Most failures were a result of connection errors associated with probes not being able to `connect()` to the server or the network being unreachable.

When splitting the failed measurements into IPv4 and IPv6 for further analysis, we find that the test reports the IP endpoint for 189k of the 282k failures. Failures that cannot be assigned to either address family are not considered in further analyses. We observe that the majority (83.8%) of these 189k failures was seen over IPv6, showing that failures were around five times more common over IPv6 in comparison with IPv4.

We define *success rate* as the number of successful iterations to the total number of identifiable iterations of the test over an address family per probe. Fig. 3 shows the boxplot of daily success rates per probe for each address family, aggregated by month. It shows that median success rates are relatively close to 100%, indicating high availability of Netflix OCAs. Probes achieve slightly lower success rates over IPv6 in the first half of the measurement period, however, the success rate over IPv6 improves toward the second half, with ∼10% probes having a lower success rate than 50% over IPv6. In particular, Oct 2016 exhibits much lower success rates than any other month in the dataset; we speculate that this was likely caused by several global outages reported during that month [33], [34], [35], [36] that affected Netflix among many other services.

B. IPv6 Preference

We measure TCP connect times toward Netflix OCA servers hosting the content. The test captures this aspect by recording the time it takes for the `connect()` system call to complete. The DNS resolution time is not taken into account in this metric. This is important to measure because applications running on dual-stacked hosts will prefer connections made over IPv6 [27]. This makes `getaddrinfo()` resolve DNS names in an order that prefers an IPv6 upgrade path. The dictated order can dramatically reduce the application's responsiveness in situations where IPv6 connectivity is broken (or bad). In fact, an attempt to connect over an IPv4 endpoint will only take place when the IPv6 connection attempt has timed out. The Happy Eyeballs (HE) algorithm [29], [32] allows applications to switch to IPv4 in such situations. The HE algorithm recommends that a host, after resolving the DNS name, tries a TCP `connect()` to the first endpoint (usually IPv6). 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 (IPv4) and start a competition to pick the TCP connection that completes first.

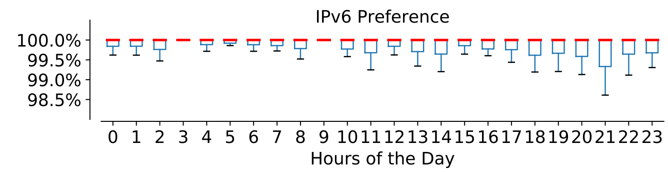

Fig. 4. Boxplot showing the IPv6 preference of TCP connections as seen each local hour of the day per probe, based on a HE timer of 250 ms. The peak usage period [37] from around 18:00–23:59 exhibits a lower IPv6 preference.

We apply the HE algorithm to the dataset to determine cases in which a probe would have preferred IPv6 over IPv4 for each *measurement pair*, i.e., the two iterations of the `netflix` test performed over IPv4 and IPv6 during one measurement cycle of a probe. For a total number of ∼970k successful IPv4 and IPv6 measurement pairs, IPv6 would have been preferred ∼964k (99.4%) of the time by the dual-stacked probes, indicating high IPv6 preference overall.

We further analyze the IPv6 preference for each probe by the local hour of the day as shown in Fig. 4. The preference of IPv6 decreases over certain peak hours (local time of probes), in particular from 18:00–23:59. This roughly aligns with the FCC definition of peak usage periods [37] as weeknights between 19:00–23:00 (local time), even though we did not consider weekdays and weekends separately in this analysis. The drop in IPv6 preference during peak hours could be reflecting the higher load that tends to increase failure rates and latency over IPv6, thus causing applications to fallback to IPv4. However, the dataset does not contain information about the load distribution to analyze this aspect in further depth.

Takeaway: Netflix OCA servers are highly available, although few probes experience lower success rate over IPv6 due to issues closer to the vantage point. Clients strongly prefer connecting to Netflix OCA servers over IPv6, however, the IPv6 preference also tends to drop during peak hours of the day.

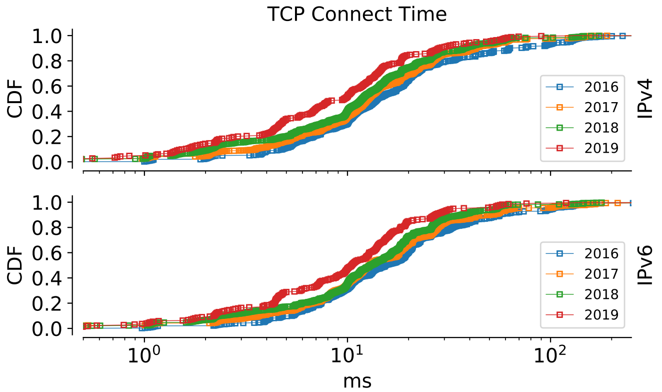

Fig. 5. Distribution of median TCP connect times toward Netflix OCAs over IPv4 (above) and IPv6 (below) per probe and day, split by years. TCP connect times have reduced from ∼25–27 ms (2016) to ∼15–17 ms (2019).

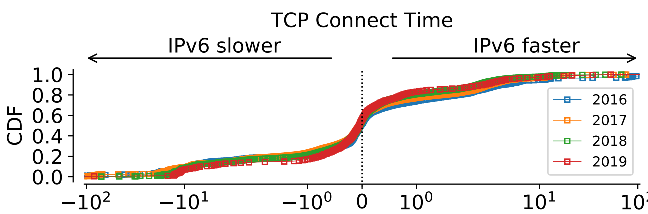

Fig. 6. Distribution of differences in TCP connect times per probe and day, split by years. Comparable TCP connect times are observed over IPv4 and IPv6, with marginal improvements in favor of IPv6 over the years.

C. Latency

We further study how the observed latency over IPv6 compares to IPv4. We observe latency by measuring the time it takes to establish a TCP connection toward Netflix OCAs and observe that TCP connect times over both address families have reduced throughout the years, as shown in Fig. 5. For instance, 75% of the IPv4 samples had TCP connect times of up to 25.2 ms in 2016. In 2017, the number improved to 22.7 ms, which denotes a relative improvement by roughly 10%. Another improvement of ∼10% was witnessed in 2018, when the TCP connect time was reduced to 20.5 ms, which was even further reduced to 15 ms in 2019 (−26.8%). For IPv6, observed changes were less pronounced in the beginning: in 2016, 75% of the samples required up to 27.7 ms for TCP connection establishment, which was only reduced to 26 ms in 2017 (−6.3%). Due to substantial reductions to 21.1 ms in 2018, i.e., about 5 ms (−19%), and to 16.4 ms in 2019 (−22.3%), IPv6 now has comparable latency to IPv4 with respect to TCP connect times. Thus, both address families showed moderate improvements of ∼40% over the years.

In order to quantify the difference in latency metrics for the measurement pairs, Fig. 6 shows the distribution of difference in TCP connect times over IPv4 and IPv6 as seen from each probe per day, split by year. The values on the positive scale indicate IPv6 being faster. It shows that no substantial changes occurred over the measurement period

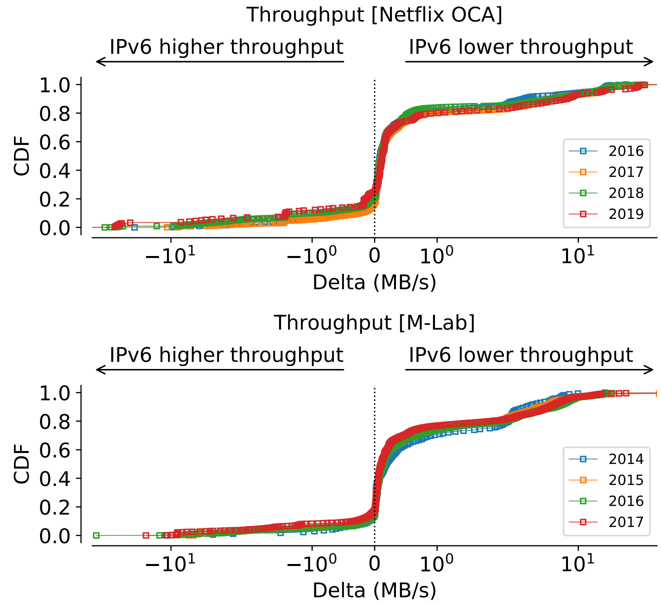

Fig. 7. Distribution of throughput differences between IPv4 and IPv6 toward Netflix OCA (above) and M-Lab (below) destinations, for each probe per day over the years. 75% of the samples exhibit lower throughput over IPv6.

regarding the differences, since the curves for each year largely overlap. Regarding TCP connect time, measurement pairs were marginally faster over IPv4; around 52% (2016) to 54% (2017, 2018), and 55% (2019) of the pairs exhibited negative deltas, meaning that around half of the measurements were faster over either address family. The cases in which IPv6 was slower than IPv4 by more than 250 ms, which represents the recommended timer of the HE algorithm [29], amounted to less than 1% in every year, corroborating high preference to download content over IPv6. We further compare the latency toward dual-stacked content caches over either address family in §V.

Takeaway: TCP connection establishment times toward Netflix OCA servers have reduced by 40% over both address families over the years. As of 2019, half of the samples connect faster over IPv6.

D. Throughput

We now investigate how the measured achieved throughput compares over both address families. Throughput toward Netflix OCA servers showed an increasing trend over both address families and over the years. For instance, 75% of the samples achieved up to 10.6 MB/s of throughput over IPv4 and 10.3 MB/s over IPv6 (2016). The throughput has then increased in the recent years, ranging from 11 and 13 MB/s for both address families. The SamKnows probes also perform `speedtest` toward Measurement Lab (M-Lab) [38] destinations, whereby we witness similar increasing trends in throughput, indicating a possible increase in line rates over the years closer to the vantage point.

Fig. 7 (above) shows the distribution of throughput deltas over IPv4 and IPv6, as seen for each probe per day since 2016

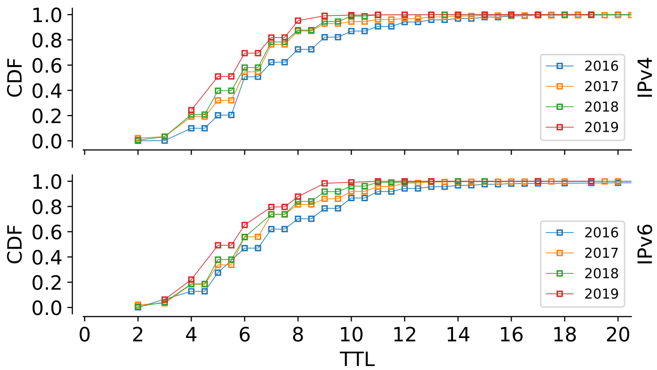

Fig. 8. Distribution of median IP path lengths toward Netflix OCAs over both address families, as seen per probe and day over the years. Netflix OCA servers are reachable in nine IP hops for most samples.

toward Netflix OCA destinations. The values on the negative scale indicate that higher throughput is achieved over IPv6. We observe that throughput over IPv6 was lower for the majority of the measurements. When directly comparing the difference in throughput for a measurement pair, IPv6 showed higher throughput only in ∼18% of the cases in 2016 (∼17% in 2017, ∼21% in 2018, ∼25% in 2019), meaning that IPv4 had higher throughput for ∼75–83% of the samples. However, ∼70–75% of the deltas were situated within −1 and +1 MB/s in each year, indicating that differences were rather small for most samples. We also investigate the throughput samples for situations where the content was downloaded from content caches over both address families, however, observed similar distributions, with the throughput over IPv4 being generally higher for caches as well. We further examine differences in throughput toward M-Lab destinations using the aforementioned `speedtest` dataset (with partially overlapping time periods), as shown in Fig. 7 (below), and again observed generally higher throughput over IPv4. As such, the lower throughput over IPv6 is not specific to Netflix destinations, indicating lower throughput over IPv6 in general.

Takeaway: The achieved throughput toward both M-Lab and Netflix OCAs has increased over the years over both address families, indicating a possible increase of line rates. IPv4 exhibits higher throughput for more than 75% of the samples in comparison with IPv6, although lower throughput over IPv6 is not specific to Netflix.

E. IP Path Lengths

We use the `scamper` dataset to examine whether IP path lengths toward Netflix destinations have improved over the observation period. In the majority of the cases (see Fig. 8), Netflix OCA servers were reachable within nine IP hops over either address family, with an increasing number of measurements exhibiting shorter paths as time progressed. For instance, this upper bound of nine IP hops covered 82.1% of the IPv4 paths and 78.5% of the IPv6 paths in 2016, which increased to a coverage of 92.8% (IPv4) and 86.1% (IPv6) in 2017. In the subsequent year (2018), 94.5% of the IPv4 and 91.8% of the IPv6 destinations were reached within nine IP hops, with close to all samples being reachable within nine IP hops in 2019 (IPv4: 99.1%, IPv6: 98.4%). This indicates that slightly more paths toward Netflix OCAs are shorter over IPv4, though both address families behave quite comparably. This observation is reflected in the previous results as well, where IPv4 was seen to perform marginally better than IPv6. Consequently, this path length analysis also reveals that Netflix OCAs have moved closer toward the edge network over both address families throughout the course of the measurement period, indicating efforts by Netflix and ISPs [39] to bring content closer to the users.

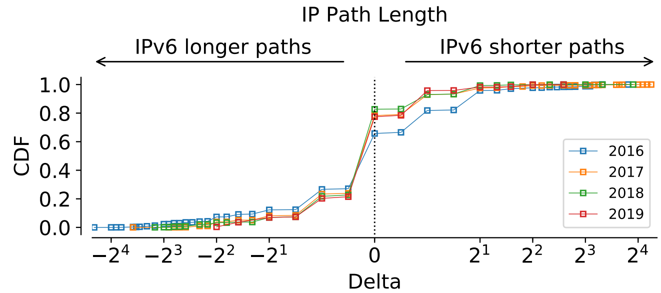

Fig. 9. Distribution of IP path length differences between IPv4 and IPv6 toward Netflix OCAs, as seen per probe and day since 2016. An increasing number of paths over IPv4 and IPv6 were even in length, as shown by 38% of the samples in 2016, compared to 56% in 2019.

Fig. 9 shows that around 27% of the samples had longer paths over IPv6 (2016), compared to 21.5% in 2019. In 2016, 33.5% of the paths were shorter over IPv6, while in 2019 this only held for 22.5% of the samples. Further, more paths over both address families have become evenly long, increasing from 38% of the measurements in 2016 to 56% in 2019. We suspect that this observation is a consequence of native IPv6 connectivity receiving higher adoption, as tunneling services which inflate IP path lengths were commonly used in the past but are progressively getting replaced by native IPv6.

Takeaway: Over the years, Netflix OCAs have moved closer to the edge network with increasing numbers of IP paths becoming evenly long. Most OCAs are reachable within nine IP hops over either address family.

V. Content Cache Deployments

Content cache deployments bring performance benefits by bringing content closer to the user [39], [40], [41]. However, these benefits have not been quantified specifically for Netflix OCA deployments in practice. Our longitudinal dataset allows us to carry out a quantitative comparison of latency and achieved throughput benefits as witnessed when the content is delivered from a Netflix OCA deployed directly within the ISP's network, in contrast to when the content is delivered from a Netflix OCA deployed outside the ISP boundary.

Toward this goal, we employ a heuristic for both address families where we map the source IP endpoint of the mea-

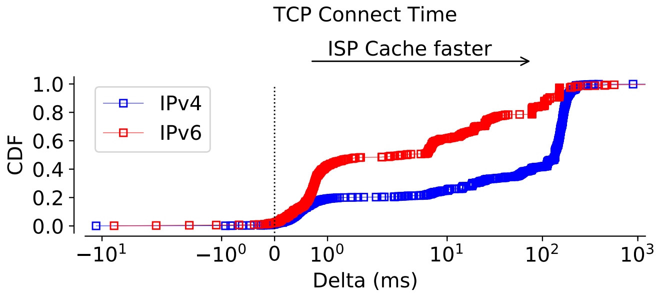

Fig. 10. Distribution of differences between Netflix CDN and ISP caches for TCP connect times over both address families. Latency toward the Netflix CDN is higher compared with latency to ISP caches for nearly all of the TCP connect time samples over both address families.

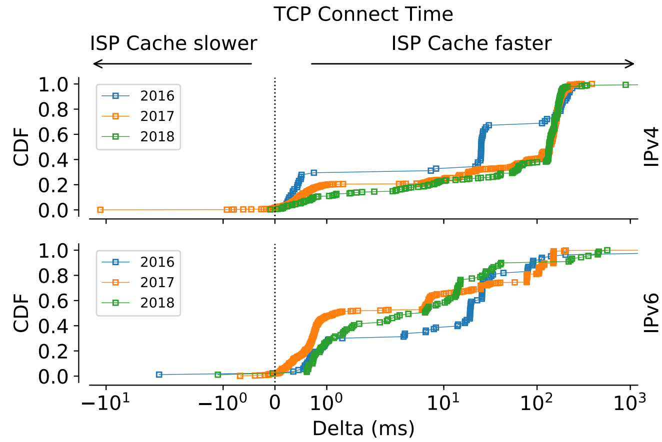

Fig. 11. Distribution of daily differences (per probe) between Netflix CDN and Netflix OCAs deployed within ISP network for TCP connect times over both address families since 2016. Netflix OCA deployed within the ISP exhibit lower latency, with differences over IPv4 becoming more and over IPv6 becoming less extreme over years.

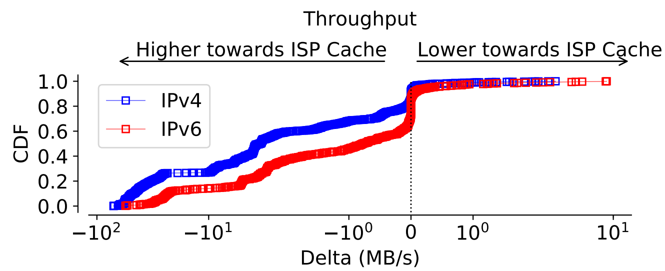

Fig. 12. Distribution of differences between Netflix CDN and ISP caches for throughput over both address families. The majority of the samples show higher throughput toward Netflix OCA caches deployed in ISP networks than toward Netflix CDN over both address families.

surement vantage point and the destination IP endpoint of the Netflix OCA to the ASes that announce the most specific encompassing IP prefix. A match in the source and destination AS numbers indicates that the request did not leave the ISP boundary, revealing the deployment of a Netflix OCA *cache* within the ISP network. Other cases are labeled as *CDN* deployments. Considering that ∼80% of the probes measure from home networks, this heuristic allows to measure characteristics for cases in which a Netflix OCA deployed within the ISP boundary could be identified with reasonable certainty. However, note that the goal is not to exhaustively identify all Netflix OCA deployments within the ISP but to understand latency and path length implications in situations where content caches are identified with reasonable certainty.

A. Latency

We begin by quantifying the benefit of deploying Netflix OCAs directly within the ISP network. We calculate the difference in TCP connect times between cache and CDN deployments. For each probe, two daily medians were calculated for each address family, one for measurements toward a cache, and one for measurements toward CDN deployments.

Fig. 10 shows the distribution of the difference between the two daily medians for both address families. Positive values indicate that latency toward the Netflix OCA deployed within the ISP network is lower compared with when the Netflix OCA is deployed in the Netflix CDN. Regarding TCP connect time, close to all of the measurements were faster toward the ISP cache, as one would expect. Caches were faster by 150 ms than the CDN servers in 61.4% (IPv4) and 92.3% (IPv6) of the cases, highlighting the importance of dual-stacking content caches in ISP networks.

Considering the longitudinal evolutions of these delta metrics, Fig. 11 shows the distribution for each year in the study period. Considering TCP connect time, ISP caches became much faster compared with the CDN servers over IPv4, depicted by the shift to the right from 2016 to 2017 and 2018. Over IPv6, ISP caches were initially much faster than CDN deployments in 2016, although latency toward the CDN has caught on in 2017 and 2018, visualized by the shift toward the left. As such, the latency implications of content delivery without caches over IPv6 has also reduced with increasing native IPv6 deployment and consequent reduction of tunnels, which used to inflate latency when crossing inter-domain links.

We suspect that the reduction in IP path length bringing caches closer to the edge helps to reduce latency; at the same time, increasing native IPv6 adoption also helps lowering the latency toward the CDN over IPv6.

B. Throughput

Quantifying the benefit in achieved throughput (Fig. 12), we witness that over IPv4, only ∼10% of the samples achieved a higher throughput toward Netflix OCAs outside of the ISP network. On the other hand, roughly 63% had a higher throughput of up to 10 MB/s more toward an OCA cache over IPv4. Regarding IPv6, ∼78% of samples had a higher throughput toward the ISP cache deployments, with ∼63% achieving up to 10 MB/s more, showing that throughput can be much higher toward an ISP cache compared with having to leave the ISP boundary to download content. For 75% of samples, the absolute throughput was increased by the presence of a cache from ∼11.1 MB/s to ∼34 MB/s over IPv4 and from ∼10.5 MB/s to ∼32.3 MB/s over IPv6. As

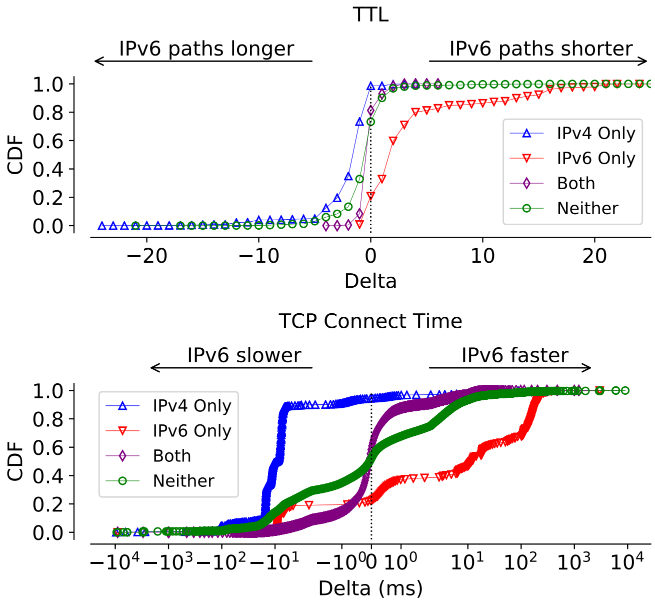

Fig. 13. Distribution of differences in IP path length (above) and TCP connect time (below), classified by cache deployments over: (a) IPv4 only; (b) IPv6 only; (c) both and (d) neither address family. In cases when a cache is available over one address family only, distributions shift away from congruency; when caches are dual-stacked, latency and IP path lengths appear comparable.

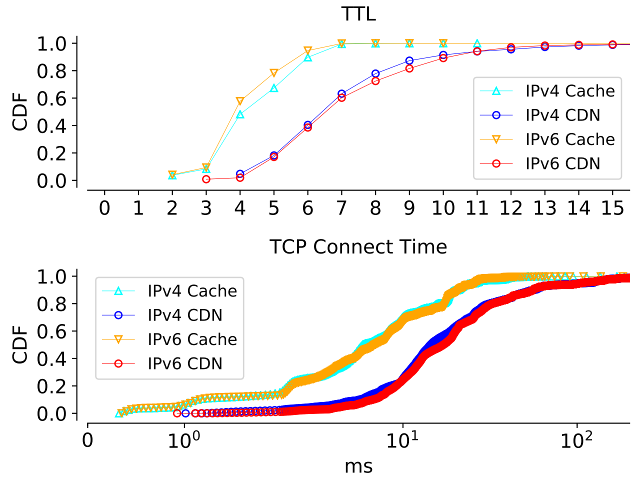

Fig. 14. Distribution of IP path lengths (above) and TCP connect times (below) for all measurement pairs, split by cache and CDN deployments. ISP caches can reduce IP path lengths by 40% over IPv4 and by 50% over IPv6. Latency improvements are also visible, with TCP connect times being reduced by up to 64% over both address families.

such, caches are able to sustain higher throughput by a factor of up to three in comparison with downloading content from a CDN server.

Overall, more samples exhibited a higher throughput toward the cache nodes over IPv4 compared with IPv6. While only ∼55% of the samples were faster toward caches by more than 1 MB/s over IPv6, 68.2% of the cases fulfilled this criterion for IPv4. Even though throughput differences were comparable, more IPv4 samples exhibited higher throughput for the cache deployment, indicating room for improvement regarding IPv6 cache deployments.

Takeaway: OCAs deployed inside the ISP boundary lower the latency over both address families. Over the years, the latency benefits toward such caches have become more pronounced over IPv4 and less pronounced over IPv6. Throughput can also increase by a factor of up to three when streaming from caches over either address family.

C. Caches by Address Family

We split IP path length and TCP connect time distributions for the measurement pairs into four categories: the content was downloaded from a) a cache over IPv4 but not over IPv6, b) a cache over IPv6 but not over IPv4, c) caches over both address families, and d) caches over neither address family.

Whenever a cache was only identified for one address family, i.e., a) and b), it should be expected to witness shorter paths and lower latency over that address family, as content would be delivered from a cache within the ISP's network. The distributions of the four cases are shown in Fig. 13, visualizing that assumption with the blue (a) and red (b) curves shifting outwards (away from zero). Importantly, the offset is less pronounced when the cache was only available over IPv6 (red) compared with the offset of IPv4 (blue). The distribution representing no identified caches over either address family (green) exhibits the largest variation with respect to all metrics. Lastly, for the situation where the content was delivered over a cache on both address families (purple), the distribution converges closer to zero (congruency), which indicates that caches perform comparably and predictably over both IPv4 and IPv6 with less variance.

When a cache was deployed over both address families, we observed that ∼72% of the samples had equal IP path lengths. Furthermore, ∼19.7% of the measurement pairs witnessed shorter paths over IPv6, indicating that IPv6 caches can be reached by the user in less number of IP hops than IPv4 caches. TCP connect times for ∼58% samples were lower over IPv4; consequently, ∼42% of the samples were faster over IPv6. This observation shows that IP path lengths do not tightly correlate with latency. For situations where caches were only available over IPv4, ∼73.4% of the measurement pairs had shorter IP paths toward the IPv4 caches compared with the IPv6 CDN servers. Samples that only had a cache available over IPv4 had lower TCP connect times over IPv4 in 94% of the cases. When the cache was identified to be on the IPv6 side only but not over IPv4, ∼79.2% of the paths were shorter over IPv6, and similarly, TCP connect times were lower for 78%. Overall, we found several cases in which content was downloaded from a cache over one address family only, meaning that not all caches are properly deployed in dual-stack operation. This indicates another area of improvements for ISPs deploying Netflix OCA boxes within their network.

In order to assess the benefits that Netflix OCAs bring when deployed inside the ISP network, we quantify the differences in IP path lengths and latencies within the same address family. The distribution of the IP path lengths and latency for all samples, split by cache and CDN, are depicted in Fig. 14, further differentiated by address family. For $\sim$90% of the samples, caches were reached within six IP hops over IPv4, whereas $\sim$91% of the CDN destinations had paths up to ten IP hops. Consequently, caches can reduce IP path lengths by up to 40% over IPv4. Considering IPv6, 95% of the samples had a path length of at most six IP hops toward caches, while the CDN required up to eleven to twelve IP hops. As such, caches reduce path lengths by up to 50% over IPv6, indicating a higher potential benefit regarding path length reduction when deploying caches over IPv6.

Regarding latency benefits, we compared the distribution of TCP connect times for cache and CDN scenarios. For instance, TCP connect time over IPv4 and IPv6 behaved much more similarly: 90% of the samples required $\sim$21 ms to reach a cache, while it took the same fraction $\sim$56–58 ms to reach the CDN over both address families. Thus, caches are able to reduce TCP connect times by up to $\sim$64%.

Overall, these observations highlight, and foremost quantify, the notable benefits that deploying caches in the edge network brings for content delivery. As such, caches within the ISP's network play a crucial role, particularly in dual-stack operation; otherwise, latency and throughput degradation tend to hamper content delivery over IPv6.

Takeaway: Latency benefits are more pronounced when caches are available over IPv4-only as opposed to IPv6-only. In situations where caches are dual-stacked, it takes less number of IP hops over IPv6 to hit the cache, although more samples still exhibit lower latency over IPv4. Consequently, IP paths do not tightly correlate with latency. Content caches are reachable within six IP hops and can reduce path lengths by 40% over IPv4 and by half over IPv6. As a result, TCP connect times are reduced by 64% over both address families.

VI. Limitations and Future Work

Limitations – We are aware that due to the distribution of SamKnows probes, observations can be biased around Europe, North America, and Japan. Yet, current IPv6 deployment efforts are also centered around these regions, however, the state of IPv6 adoption may be different in the future.

The identification of caches is limited by the heuristic of matching source and destination ASNs, which allows identification of ISP caches with reasonable certainty. In contrast, caches located within the ISP network but with IP address prefixes announced by Netflix ASes cannot be reliably distinguished from non-cache CDN deployments. However, recall that the goal is not to exhaustively identify all caches but to understand latency and path length benefits of ISP caches.

Future Directions – In situations where the OCA is deployed outside the ISP boundary, peering between the interdomain links starts to play a role. Particularly, when such links are under-provisioned, increased traffic flow can lead to congestion. In the future, we plan to investigate how Netflix content delivery deteriorates in situations of recurring congestion [7] at interdomain links. We further identify a more in-depth analysis as future work, which includes failures as well as differentiation between ASes and geographical regions.

VII. Conclusion

Using a longitudinal dataset collected by deploying `netflix` test on $\sim$100 dual-stacked SamKnows probes, we studied Netflix content delivery from multiple perspectives. We observed that the Netflix OCAs are highly available, exhibiting a high success rate of establishing connections. We observed that it used to take $\sim$25–27 ms (2016) to establish a TCP connection toward Netflix OCAs, which has reduced to $\sim$15–16 ms (2019), showing similar latencies over both address families. We saw that Netflix OCAs can be reached within nine IP hops. This quantification helps providing evidence on how popular content is getting pushed to the edge and closer to the users over time. When comparing content delivery over different address families, we witnessed that due to comparable TCP connect times, clients strongly prefer connecting to the Netflix OCAs over IPv6. However, IPv6 preference drops over certain peak hours during the day. Nevertheless, the achieved throughput is lower over IPv6, which may lead to degraded user experience, although the observation is not specific to Netflix exclusively. ISPs can employ multiple strategies to stream content from Netflix OCA deployments. We observed that Netflix OCAs that are integrated in the ISP's AS tend to reduce TCP connect times by $\sim$35 ms ($-$64%). The achieved throughput can also increase by a factor of three. We observed that such ISP caches are reachable within six IP hops. We hope our empirical data will allow simulation studies to appropriately model Netflix content delivery in future studies. We also recommend ISPs to embed OCAs directly within the ISP network to ensure lower latency and higher throughput over both address families. The results presented in this paper highlight the importance of data collection over a longitudinal period to understand the evolution of content delivery of a popular video streaming service on the Internet, particularly in light of IPv6 adoption and increasing edge deployments.

Reproducibility Considerations – We release an anonymized version of the longitudinal datasets along with the analysis scripts[1] to allow further analysis and exploration of the measurements.

Acknowledgment

We thank Jörg Ott and the anonymous reviewers for their valuable feedback on the manuscript. We also thank Abhishek Kapoor for helping with an early exploration of the dataset.

References

[1] Sandvine, "Global Internet Phenomena (Latin America and North America)," June 2016, [accessed 2020-Jan-08]. [Online]. Available: https://goo.gl/LvzcZf

[1] https://github.com/tv-doan/infocom-2020-netflix

[2] Victor Luckerson, "Netflix Accounts for More Than a Third of All Internet Traffic," May 2015, [accessed 2020-Jan-08]. [Online]. Available: http://time.com/3901378/netflix-internet-traffic
[3] N. Bargisen, "Netflix and IPv6," Nov 2017, [accessed 2020-Jan-08]. [Online]. Available: https://www.ripe.net/participate/meetings/regional-meetings/ipv6-day-denmark
[4] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem Through the Lens of the Netflix CDN," *SIGCOMM CCR*, 2018. [Online]. Available: http://doi.acm.org/10.1145/3211852.3211857
[5] P. Gill, M. F. Arlitt, Z. Li, and A. Mahanti, "The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse?" in *PAM*, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-79232-1_1
[6] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Böttger, and W. Willinger, "Peering at Peerings: On the Role of IXP Route Servers," in *IMC*. [Online]. Available: http://doi.acm.org/10.1145/2663716.2663757
[7] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. J. Luckie, R. K. P. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and kc claffy, "Inferring Persistent Interdomain Congestion," in *SIGCOMM*, 2018. [Online]. Available: https://doi.org/10.1145/3230543.3230549
[8] V. Bajpai and J. Schönwälder, "A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts," *Communications Surveys and Tutorials*, 2015. [Online]. Available: https://doi.org/10.1109/COMST.2015.2418435
[9] M. J. Luckie, "Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet," in *IMC*, 2010. [Online]. Available: https://doi.org/10.1145/1879141.1879171
[10] V. Bajpai, A. Brunström, 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," *SIGCOMM CCR*, 2019. [Online]. Available: https://doi.org/10.1145/3314212.3314217
[11] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. Zhang, "Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery," in *INFOCOM*, 2012. [Online]. Available: https://doi.org/10.1109/INFCOM.2012.6195531
[12] V. K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z. Zhang, M. Varvello, and M. Steiner, "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," *Transactions on Networking*, 2015. [Online]. Available: https://doi.org/10.1109/TNET.2014.2354262
[13] J. Martin, Y. Fu, N. Wourms, and T. Shaw, "Characterizing Netflix Bandwidth Consumption," in *CCNC*, 2013. [Online]. Available: https://doi.org/10.1109/CCNC.2013.6488451
[14] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard," in *IMC*, 2012. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398800
[15] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *SIGCOMM*, 2014. [Online]. Available: https://doi.org/10.1145/2619239.2626296
[16] T. Böttger, F. Cuadrado, and S. Uhlig, "Looking for Hypergiants in PeeringDB," *SIGCOMM CCR*, 2018. [Online]. Available: https://doi.org/10.1145/3276799.3276801
[17] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger, "There is More to IXPs than Meets the Eye," *SIGCOMM CCR*, 2013. [Online]. Available: https://doi.org/10.1145/2541468.2541473
[18] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a Large European IXP," in *SIGCOMM*, 2012. [Online]. Available: https://doi.org/10.1145/2342356.2342393
[19] T. V. Doan, L. Pajevic, V. Bajpai, and J. Ott, "Tracing the Path to YouTube: A Quantification of Path Lengths and Latencies Toward Content Caches," *Communications Magazine*, 2019. [Online]. Available: https://doi.org/10.1109/MCOM.2018.1800132
[20] S. Fedorov and E. Livengood, "Building fast.com," Aug 2016, [accessed 2020-Jan-08]. [Online]. Available: https://medium.com/netflix-techblog/building-fast-com-4857fe0f8adb
[21] B. Augustin, T. Friedman, and R. Teixeira, "Measuring Load-balanced Paths in the Internet," in *IMC*, 2007. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298329
[22] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband Internet Performance: A View From the Gateway," in *SIGCOMM*, 2011. [Online]. Available: https://doi.org/10.1145/2018436.2018452
[23] Z. S. Bischof, F. E. Bustamante, and R. Stanojevic, "Need, Want, Can Afford: Broadband Markets and the Behavior of Users," in *IMC*, 2014. [Online]. Available: https://doi.org/10.1145/2663716.2663753
[24] J. P. Rula, Z. S. Bischof, and F. E. Bustamante, "Second Chance: Understanding Diversity in Broadband Access Network Performance," in *SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*, 2015. [Online]. Available: https://doi.org/10.1145/2787394.2787400
[25] Z. S. Bischof, F. E. Bustamante, and R. Stanojevic, "The Utility Argument - Making a Case for Broadband SLAs," in *PAM*, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-54328-4_12
[26] P. Richter, M. Allman, R. Bush, and V. Paxson, "A Primer on IPv4 Scarcity," *SIGCOMM CCR*, 2015. [Online]. Available: https://doi.org/10.1145/2766330.2766335
[27] D. Thaler, R. Draves, A. Matsumoto, and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)," *RFC*, vol. 6724, 2012. [Online]. Available: https://doi.org/10.17487/RFC6724
[28] D. Wing and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts," *RFC*, vol. 6555, 2012. [Online]. Available: https://doi.org/10.17487/RFC6555
[29] D. Schinazi and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency," *RFC*, vol. 8305, 2017. [Online]. Available: https://doi.org/10.17487/RFC8305
[30] Internet Society (ISOC), "State of IPv6 Deployment," May 2017, [accessed 2020-Jan-08]. [Online]. Available: https://www.internetsociety.org/resources/doc/2017/state-of-ipv6-deployment-2017
[31] Google, "IPv6 Adoption Statistics," [accessed 2020-Jan-08]. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html
[32] V. Bajpai and J. Schönwälder, "A Longitudinal View of Dual-Stacked Websites - Failures, Latency and Happy Eyeballs," *Transactions on Networking*, 2019. [Online]. Available: https://doi.org/10.1109/TNET.2019.2895165
[33] A. Abhishta, R. van Rijswijk-Deij, and L. J. M. Nieuwenhuis, "Measuring the Impact of a Successful DDoS Attack on the Customer Behaviour of Managed DNS Service Providers," *SIGCOMM CCR*, 2018. [Online]. Available: https://doi.org/10.1145/3310165.3310175
[34] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *USENIX Security Symposium*, 2017. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
[35] E. Weise, "Netflix down for about 2.5 hours Saturday," Oct 2016, [accessed 2020-Jan-08]. [Online]. Available: https://www.usatoday.com/story/tech/news/2016/10/01/netflix-goes-down-saturday-afternoon/91396200/
[36] N. Woolf, "DDoS attack that disrupted Internet was largest of its kind in history, experts say," Oct 2016, [accessed 2020-Jan-08]. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
[37] Federal Communications Commission (FCC), "Measuring Broadband America Report - 2016," Dec 2016, [accessed 2020-Jan-08]. [Online]. Available: https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-fixed-broadband-report-2016
[38] C. Dovrolis, P. K. Gummadi, A. Kuzmanovic, and S. D. Meinrath, "Measurement Lab: Overview and an Invitation to the Research Community," *SIGCOMM CCR*, 2010. [Online]. Available: https://doi.org/10.1145/1823844.1823853
[39] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. M. Maggs, J. Rake, S. Uhlig, and R. Weber, "Pushing CDN-ISP Collaboration to the Limit," *SIGCOMM CCR*, 2013. [Online]. Available: https://doi.org/10.1145/2500098.2500103
[40] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann, "Improving Content Delivery Using Provider-aided Distance Information," in *IMC*, 2010. [Online]. Available: https://doi.org/10.1145/1879141.1879145
[41] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-Aware Caching and Traffic Management in Content Distribution Networks," in *INFOCOM*, 2011. [Online]. Available: https://doi.org/10.1109/INFCOM.2011.5935123