On the Constancy of Latency at the Internet's Edge

Aditya Bhat
BITS Pilani, Goa
Zuarinagar, Goa, India
f20212071@goa.bits-pilani.ac.in

Vaibhav Ganatra
BITS Pilani, Goa
Zuarinagar, Goa, India
f20190010g@alumni.bits-pilani.ac.in

Aniket Shaha
BITS Pilani, Goa
Zuarinagar, Goa, India
f20190463g@alumni.bits-pilani.ac.in

Balakrishnan Chandrasekaran
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
b.chandrasekaran@vu.nl

Vinayak Naik
BITS Pilani, Goa
Zuarinagar, Goa, India
vinayak@goa.bits-pilani.ac.in

***Abstract*—Latency is a key determinant of application performance and, most importantly, end-user experience. While a low latency value between any pair of endpoints will immensely benefit the applications or data transfers between those endpoints, whether that end-to-end latency fluctuates and, if yes, to what extent also have crucial performance implications. Recently, Davisson et al. replicated a decade-old seminal work on assessing the constancy of latency between arbitrary endpoints in the Internet. In this work, we follow the approach of Davisson, albeit with a focus on end users: We assess the constancy of latency between end users and the servers that serve most of the content the end users typically consume. These are the endpoints that are of interest. We measure latencies between RIPE Atlas anchors and the edge servers of various content delivery networks (CDNs), both selected from diverse geographic locations and networks. We show that end-to-end latencies between end users and CDN edge servers are stable, with few significant spikes or variations. We find that the durations without change for the analyzed paths are much larger, with median maximum values at least three times greater than those in previous studies on RIPE Atlas anchor paths. We release our tools and analyses to foster open and reproducible research.**

***Index Terms*—Latency, Content delivery networks, Network measurements**

I. Introduction

Latency dictates the performance of applications and, most importantly, end-user experiences. According to Google, an increase of $100\,\mathrm{ms}$ reduces the daily searches per user by 0.2%, which is significant since the searches per user is an effective proxy for user satisfaction [1]. Similarly, a Facebook study found that network latencies are typically the bottleneck in optimizing the performance of large-scale Internet services [2]. A recent study on virtual reality (VR) streaming showed that reducing the latencies between the client and the server has substantial implications for bandwidth requirements [3]. A rich body of prior work focused, hence, on measuring network latencies [4], [5], [6], [7], analyzing how they impact applications' performances [8], [9], [10], [11], [12], and investigating methods to reduce latencies [13], [14], [15], [16].

Although reducing network latency immensely benefits all kinds of applications, whether end-to-end latencies are *stable* or *constant* over a period of time has substantial performance implications. For example, TCP throughput can severely degrade when it experiences substantial fluctuations in round-trip times (RTTs) [17], [18]. Latency fluctuations, especially non-congestive jitter, can introduce starvation with delay-based congestion control algorithms (CCAs) [19]. A recent study shows that even small changes in latency have fairness implications in modern CCAs [20]. Therefore, previous work has analyzed latency variations under three different notions of *constancy*. The first such work was performed about a quarter century ago by Zhang et al. [21]. Four years ago, Davisson et al. reassessed the constancy of latency given that the Internet has evolved dramatically since that original study [22]. Although these two prior works looked at end-to-end latencies, i.e. RTTs, across arbitrary network paths on the Internet, we focus on latencies of paths between end users and the servers from which they typically consume data.

Today, content delivery networks (CDNs) deliver a substantial fraction of the content consumed by end users [23]. If the end-to-end latencies of the paths between the CDN servers and end users are *constant*, it will greatly benefit the delivery of content to end users; as a consequence, the end users' quality of experiences (QoEs) would be high. We, therefore, apply the methodology of Davisson et al. [22], but tailor and extend the approach to characterize the latencies between CDN servers and end users. We use RIPE Atlas anchors as proxies for end users in diverse networks, as well as geographic locations, and identify the "edge" servers of five widely used CDNs as targets for our latency measurements. CDNs were not well-known or widely used during the study of Zhang et al. [21]; indeed Akamai, one of the largest CDNs, gained prominence only towards the end of 2001. Davisson et al. [22] also do not use CDN servers in their replication.

In this work, we replicate the measurement study of Davisson et al. [22] to assess how constant (i.e., free from variations) end-to-end RTTs are in the Internet. Unlike this prior work, we focus on the RTTs of the paths between CDN servers and end-user hosts (§III). Due to the infrastructure limitations, we gathered our measurements once every two hours instead of at every four minutes as in prior work, although our targets

978-3-903176-74-4

are different and diverse than those in prior work. To this end, we fetched the web pages of "popular" websites (from the Hispar list [24]) and identified 100 CDN edge servers used in serving the content on these web pages. These edge servers, belonging to the top five widely used CDNs, served as the targets in our study. We then chose 69 RIPE Atlas anchors as the vantage points. In the prior work, RIPE Atlas anchors served both as targets and vantage points [22]. Our replication effort, hence, offers a natural extension of this prior work [22]: Whereas prior work studied the constancy of latency between arbitrary Atlas anchors, we focus on the latency between such anchors and CDN edge servers. We, then, measured the latency between our vantage points and targets to assess their mathematical (§IV-B), operational (§IV-C), and predictive (§IV-D) constancy.

We summarize our contributions as follows.

⋆ We replicate the prior work ([21] and [22]) on assessing the constancy of latency in the Internet by measuring and analyzing the end-to-end latencies of paths between CDN edge servers and end users.

⋆ We show empirically that latencies of paths between end users and CDN edge servers are largely devoid of significant latency spikes, as well as variations. Prior work reports that about 0.01% of the observations experience latency spikes that are a factor of 10 or higher than the median of the corresponding timelines. Our work, in contrast, reveals that 3.84% of the latency observations, of paths between end users and CDN servers, experience such significant latency spikes. About 7.5% of the latency observations deviate by at least a factor of 2 or higher from the median observed value (in corresponding timelines). Therefore, though latency spikes are not typical, in the atypical cases where we observe them, they are quite significant.

⋆ We observe that the time durations for which the end-to-end path latencies remained constant in our study are larger than those reported in prior work. The median time spans for which we observe latencies to be (mathematically) constant are at least three times larger than the corresponding values from prior work. Our measurements also indicate that the median and maximum values of such change-free time periods of Intra-AS paths are smaller than those of inter-AS paths, but such changes do not affect the path RTTs adversely.

⋆ We compare the observed end-to-end latencies to the theoretically achievable minimums and show that a small fraction of observations are substantially larger than what the theoretical minimums.

⋆ We release our tools and analyses as open-source artifacts on GitHub [25].

Ethics. *This work does not raise any ethical issues.*

II. Related work

End-to-end latencies of the network path between any two endpoints have huge implications for the performance of applications. There is, hence, a rich body of prior work on characterizing the latencies between arbitrary endpoints on the Internet. The measurement studies conducted by Vern Paxson are likely the earliest large-scale studies on this topic (e.g., [26]). Paxson's efforts characterized the stability of end-to-end paths (i.e., how likely were they to change over time?) and analyzed their impact on latency. Chandrasekaran et al. replicated this study using vantage points of a large CDN and showed that most paths in the core of the Internet are typically stable and changes typically do not introduce substantial changes in end-to-end latency [27]. The scope of these studies was on the path-level stability and its impact on latency. Our study, in contrast, focuses on the stability of the latencies themselves. Appel et al.[28] emphasize the importance of using a diverse set of vantage points for network measurement, and our vantage point and target selection approaches follow the recommendations of this study.

Several recent studies have examined latency inflations and their implications for the performance of networked systems and applications [5], [4]. Singla et al. first identified infrastructural inefficiencies as the cause of latency inflation [5]. Building on this, Bozkurt et al. showed that reducing the Internet's infrastructural inefficiencies instead of optimizing protocol overhead is the key to reducing latency inflation [4]. While we characterize the latency inflations between our vantage points and targets, this work mostly focuses on the stability or constancy of the observed latencies.

Martin et al., in their recent study, showed that cloud providers reduced latencies via edge data centers, although this approach created a 'cloud digital divide,' where users in close proximity to the data centers experienced a much lower latency than those that were farther away—they showed how this divide disproportionately affected lower-income regions [29]. Ingabire et al. measured public cloud latency with large-scale RIPE Atlas measurements, showing that while distance was the major cause of latency, network stability was high [30]. Bose et al. showed that Starlink users faced high CDN latencies and proposed SpaceCDNs to reduce this latency by more than 50% [31]. This work does not investigate the factors that contribute to latency, but on whether the observed latencies are stable over time.

To the best of our knowledge, the earliest work analyzing the constancy of end-to-end latencies in the Internet was from Zhang et al. [21]. Recently, Davisson et al. reproduced this work in light of the substantial topology and technology changes since the first study [22]. We replicate the work of Davisson et al. but focus on the end-to-end latencies of paths between end users and CDN edge servers—the path that end users might typically use to consume most of their content.

III. Background & Methodology

We characterize the end-to-end latencies of the paths over which end users predominantly consume their data, and analyze whether these latencies are constant in terms of *mathematical*, *operational*, and *predictive* notions of constancy. Per the *mathematical* notion of constancy, we consider the measured latencies between two endpoints as constant if they do not exhibit any "level shifts" in latency. Said differently, we set aside *transient* variations in latencies and analyze whether the

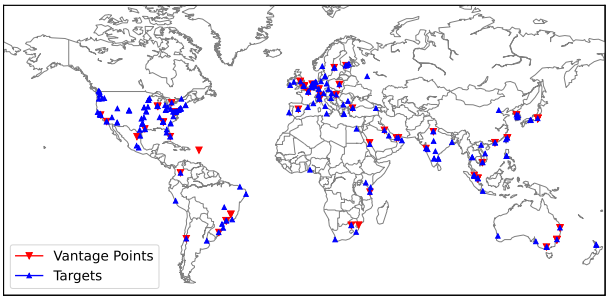


Fig. 1: *Geographic locations of vantage points from the RIPE Atlas platform and targets (i.e., CDN servers).*

baseline latency (modulo the variations) exhibits a substantial shift in value. This notion of a level shift in latency and inferring them using changepoint detection algorithms is quite common in the literature (e.g., [26], [21], [27], [22]). The *operational* notion of constancy, in contrast to the mathematical definition, simply defines latencies as constant if the variations they experience are within a small finite range. A network operator, for instance, may consider latency variations within a range of 20 ms to be simply an artifact of transient issues or even measurement noise; only when the latencies exceed this operational threshold, they may consider taking action to optimize the latencies. The last, *predictive* notion of constancy simply deems latencies as constant if they are predictable: If the error in our forecast or prediction of a latency between two endpoints (based on historical data) is within a margin of tolerance, then we deem the latencies as predictively constant.

In this work, we measured the latencies (i.e., `pings`) between vantage points that serve as proxies to end users and the edge servers of major CDNs over a period of ten days using the RIPE Atlas infrastructure. Although the volume of data traversing the reverse path (i.e., from an edge server to an end user) is typically much larger than that along the forward path (i.e., from the end user to the edge server), the latencies of both the paths have implications for the end user applications' performance. Below, we explain how we select the vantage points and targets in our study and describe how we gather latency measurements as well as analyze them.

A. Vantage points

We used the RIPE Atlas platform to gather measurements. We used their well-documented REST APIs to orchestrate measurement campaigns and collect ping measurements between RIPE Atlas anchors and CDN edge servers [32]. The Atlas infrastructure allows two types of nodes or vantage points: *probes* and *anchors*. Probes are small, dedicated hardware devices distributed worldwide to volunteers and organizations interested in participating in Internet measurement activities [33]. Anchors, in contrast, are powerful and *stable* Internet measurement devices and typically less susceptible to interference [34]. Unlike probes, volunteers do not typically host anchors but are maintained by the RIPE NCC and other collaborating organizations, e.g., universities and academic institutions. Anchors, therefore, may have higher bandwidth

TABLE I: *Characteristics of vantage points and targets*

Measure or Description	***Vantage Points***	***Targets***
#anchors or (target) domains	79	100
Unique IP-addresses	69	13,934
Unique /24 Networks	69	4545
#countries	34	56
Unique locations	74	222
Number of ASNs	27	53

Internet connectivity than probes, but this study focuses on characterizing latency, not bandwidth. The latencies observed from anchors may at times be smaller and more stable than those observed from probes, but it is hard to disambiguate the source of latency variations observed from the probes [34]. A potential future work may gather measurements from both anchors and probes to the same targets to determine whether the observed variations, if any, likely arise only in the last mile. As such, the findings from this work can serve as an optimistic lower bound on latencies that we can achieve from the present infrastructure, which has substantial implications for applications and is of huge interest to the networking community (e.g., refer [5], [4], [31], [6]).

In this study, we chose 79 anchors (deployed in each of the top 40 countries, ranked in order based on the number of anchors hosted per country) from the RIPE Atlas infrastructure. We filtered out anchors with an *uptime* of less than 25 months (about 25%) to ensure that the selected anchors are active and likely to respond to our probes. The selected anchors represent vantage points in 34 countries (Fig. 1), with about half distributed in the USA, Australia, the United Kingdom, Japan, Brazil, and Singapore. In terms of network diversity, the selected anchors are from 27 different ASNs, with the top (in terms of the number of anchors) being AS3491 (PCCW Global), AS199524 (GCORE), AS20473 (The Constant Company), AS12008 (Vercara), and AS15133 (Edgecast). Furthermore, to perform comparative analyses of the latencies of anchors in different networks in the same geography, we chose additional anchors from different networks in the same country.

B. Targets

Today, a substantial fraction of the content to end users is delivered by CDNs [23], [35]. The end-to-end latencies or RTTs of the network paths between the CDN's (*edge*) servers and end users have, hence, huge implications for the performance of the transport protocol and, as a consequence, end-user applications [17], [16]. We selected the edge servers of widely used CDNs as targets for our measurements from the RIPE Atlas anchors. We obtained the CDN edge servers by identifying the CDN-associated domain names in the URLs of objects on diverse web pages and resolving them to find their IP addresses. Although our target-selection procedure focuses on CDNs used in delivering *web* content, the infrastructure is likely shared for delivering various applications or content types today.

Selecting diverse web pages. We sampled a set of 200 URLs comprising an equal number of landing and internal

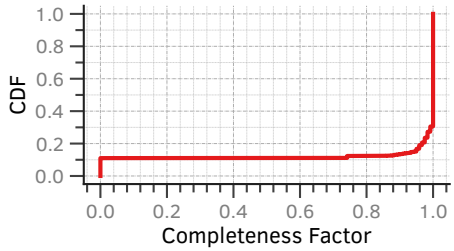


Fig. 2: *About 70% of the timelines are complete, with all 120 measurements, and 10% are void, with no latency data*

pages from the Hispar list [24]. This data set provides a rank-ordered list of 2000 websites with one landing page and 50 internal pages per website. The rank of a website captures its "popularity," i.e., how often users visit that website. First, we picked both the top 20 and the bottom 20 websites from this data set. We then sampled, uniformly at random, $\{10, 20, 30\}$ URLs from the websites in the rank-based intervals $(20, 100]$, $(100, 1000]$, and $(1000, 1980]$, respectively. For each of these 100 sampled landing pages, we also sampled one internal page, uniformly at random, from the internal pages associated with each website in the data set. This selection procedure, therefore, captures the most popular and least popular web pages. Analyzing the performance of the CDN infrastructure used to serve these diverse web pages should offer us insights into the Internet's web-serving infrastructure.

Identifying CDN servers. We crawled these web pages using the Chrome web browser launched in headless mode. We automated the web page crawls using Selenium [36] and generated an HTTP archive (HAR) file for each web-page fetch. Given a HAR file corresponding to a web-page fetch, we can obtain the various objects on that page and the URL from which each was served. We extracted these object-level URLs and used a domain-to-CDN mapping utility in a prior work [24] for identifying the servers associated with various CDNs. Since the process of determining the domains of various CDNs is based on a set of heuristics, it might occasionally fail. In these cases, we performed DNS resolutions on the domain and looked up the `whois` information on the IP addresses associated with the domain to identify the CDN. When storing the various CDN domain names, we recorded the highest and lowest ranks of the web pages where we discovered them. Then, we sampled 100 CDN domain names such that they sufficiently represented the wide range of web pages we crawled. To this end, we sampled, uniformly at random, 100 domain names, following the sampling procedure that we used for selecting diverse web pages. We restricted our analyses to the five widely used CDNs, namely Akamai, Amazon Cloudfront, Cloudflare, Fastly, and Google, which, taken together, accounted for the majority of all the CDN domains that we identified. We chose 100 targets (or CDN domain names) for our measurement campaigns, with 20 belonging to each of these five CDNs.

Per Tab. I, the vantage points (i.e., CDN domain names) that we picked resolved to 13,934 IP addresses. Before measuring the latencies, we configured our measurement campaigns to resolve the domain names from the vantage points. We geolocalized these IP addresses using the MaxMind [37] and IPInfo [38] geolocation databases. The predicted locations of IP addresses from both databases were consistent at the country level, except for a few different IP addresses where MaxMind could not geolocalize the CDN servers. The answers from either database were identical with respect to ASN. The locations of the target IP addresses, as expected, represented a diverse network as well as geographic deployment (refer Fig. 1). The target IP addresses, for instance, correspond to about 4545 unique /24 networks, which is two orders of magnitude more than that corresponding to the vantage points. They are also in nearly twice as many ASNs and deployed in 60% more countries than the vantage points.

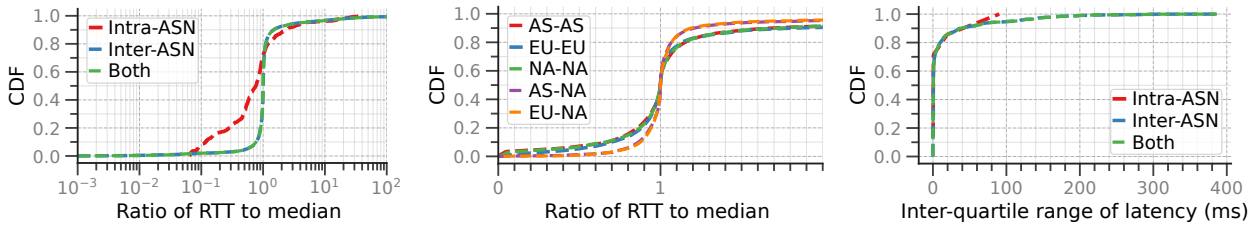
C. RTT measurements

We divided the 79 vantage points into four campaigns. Each campaign involved about 20 vantage points gathering `ping` measurements to the 100 targets. Through the four measurements *campaigns*, we gathered RTTs between our vantage points (i.e., RIPE Atlas anchors) and targets (i.e., CDN edge servers). We repeated the campaigns every 2 hours for 10 consecutive days, including 2 weekends, from March 2nd, 2024. We ran each measurement campaign using a different (RIPE Atlas) account because of the limits on the number of concurrent user-defined measurements a user can run [39].

The 120 data points, each spaced 2 hours apart over a span of 10 days, between any pair of vantage points and target constitutes a *timeline*. Of the 79 anchors used in our campaigns, 10 did not respond; the RIPE Atlas infrastructure reported a status of *abandoned* for 6 anchors and *disconnected* for 4 anchors, yielding 69 responsive anchors, which we used for collecting measurements. Each campaign had one or two unresponsive anchors, most of which were either deployed in AS12008 or in (some ASN in) Australia. About 40% of these unresponsive anchors came online only recently (i.e., a few months ago), suggesting that their offline statuses could be due to infrastructure issues. Our data set contains, hence, $69 \times 100 = 6900$ timelines, with most of the timelines being *complete*—containing all 120 data points (Fig. 2). Each data point of a timeline is the minimum latency (i.e., RTT) value observed across three probes that RIPE Atlas API sends for each `ping` measurement.

D. Characterizing constancy

To measure the constancy of latencies from a mathematical perspective (§IV-C), we used three different changepoint detection algorithms: *Bootstrap*, *RankOrder*, and the *Hidden Markov model-Hierarchical Dirichlet Process (HMM-HDP)*. These are the same methods used in prior work [22]. Bootstrap and Rank Order methods (originally used in [21]) are statistical methods to detect changes in the median; they first identify a candidate changepoint, from a timeline of latencies, and then apply a statistical test to determine whether it is significant. The HMM-HDP method is a non-parametric Bayesian model



(a) *Spikes in intra-ASN and inter-ASN paths* (b) *Spikes in intra- and inter-continental paths* (c) *Variations in latency*

Fig. 3: *(a) Latency spikes are uncommon, but 3.84% of the timelines experience spikes of substantial magnitude; (b) RTT to median ratios of intra-continental paths are higher than those of inter-continental paths—Asia (AS), Europe (EU), and North America (NA) denote the continents which contain the vantage point or the target in a vantage point-target path; and (c) Most timelines experience little to no variations in latency, although a small fraction of timelines show variations of at least 50 ms.*

that has an infinite number of states and is time dependent that makes it flexible enough to adapt to diverse time series [40]. Davisson et al. observed manually that the HMM-HDP method is more precise that the Rank Order and the Bootstrap method, that is, it detects fewer erroneous change points. We re-implemented the first two algorithms in Python based on their descriptions from Zhang et al. [21]. Davisson et al. used the implementation of the HMM-HDP algorithm provided by the RIPE Atlas platform [22], but that implementation has since been deprecated and removed. Instead, we found an open-source algorithm implementation in Julia [41]. We manually verified the implementation's accuracy using some sample timelines of (real and synthetic) latency observations and integrated it with the rest of our Python code for analyses. For assessing the operational constancy of latency (§IV-C), we simply discard variations using different thresholds and check whether the resulting latency timeline is constant. We used three methods for quantifying the predictive constancy of latencies (§IV-D). They are the simple moving average (SMA), exponentially weighted moving average (EWMA), and *Kalman* filter; these methods "smoothen" short-term latency variations in a timeline and help predict latencies in the future. The SMA uses an unweighted mean of the past W observations, providing a simple and interpretable method to smoothen timelines. The EWMA, in contrast to SMA, weights older observations exponentially based on a smoothing factor α. Since it weights recent observations more strongly than older ones, it quickly adapts to short-term trends in latencies. The Kalman filter is a recursive approach to predict the true state of a system from a series of noisy measurements. We use a simplified 1D Kalman filter with P as process variance (how stable you expect the real timeline to be) and Q as measurement variance (how noisy you expect the measurements to be).

IV. The State of the Internet's Edge

We now present the insights from our empirical observations of the end-to-end RTTs between the RIPE Atlas anchors and edge servers of various CDN servers. First, we discuss the spikes and variations in RTTs of various timelines. Then, we analyze the timelines in terms of the three notions of constancy. Lastly, we quantify how much the observed RTTs deviates from the theoretical minimum we can achieve between any pair of the endpoints in our study.

A. Latency spikes and variations

We begin by characterizing the latency *spikes* in our data set. We calculate the median RTT for each timeline (i.e., each distinct vantage point-target pair) and normalized each RTT observation for that timeline by its median, similar to prior work [21], [22]. The CDF of these ratios of normalized RTTs across all timelines (Fig. 3a) reveals that only 3.84% of the latency observations indicate spikes of large magnitude, i.e., latencies that are a factor of 10 larger than the median latency of the corresponding timeline. The prior work, in contrast, reports only 0.01% of the latency observations exhibiting such significant spikes [22]. If we consider, for each timeline, a factor of 2 or larger latency deviations from the timeline's median as a *significant* spike, only 7.5% of the observations indicate significant spikes in latencies.

The latency spikes are *not* distributed uniformly across the different networks (of vantage points and targets) in our study but focused on a small number of networks. For this analysis, we first identified the ASNs associated with the endpoints in our study; for the vantage points, we retrieved this information using the RIPE Atlas APIs. If we group the timelines based on the source ASNs (i.e., ASNs of vantage points), we observe that a few ASNs are responsible for most spikes. Two of the largest contributors are AS3491 and AS202422, each contributing about 30% of the observed latency spikes; the former is a tier-1 AS with about 700 peers and nearly 16000 ASes in its customer cone, while the latter is a small "eyeball" AS (in Luxembourg) that announces only a handful of IPv4 prefixes. Among the destination ASNs, we found four of the five CDNs we selected in our study (Cloudflare was the *exception*) appearing frequently as the destination ASN of a timeline with a substantial latency spike. These latency spikes were consistent across all days—both weekdays and weekends—in our study period.

We observed most latency spikes on inter-ASN paths, i.e., where the endpoints belonged to different ASNs (Fig. 3a). The largest latency spikes in the intra-ASN paths are about

a factor of three or lower than those observed in the inter-ASN paths. Still, most timelines in our study constitute the inter-ASN path. Also, intra-continental timelines have more latency spikes than inter continental spikes (Fig. 3b). The median values of RTTs for the latter are higher than those of the former; smaller deviations in the RTTs observed in intra-continental timelines could, hence, contribute to a latency spike. A substantial number of latency spikes, nearly two-thirds of the total, were on timelines with the destination (or target) being in North America, irrespective of their ASNs. A third of these timelines have the source (or vantage point) in North America; these intra-continental paths (i.e., between endpoints in North America) account for about 21% of the overall latency spikes observed across all timelines.

Our measurements indicate that on the paths between end users and CDN edge servers, latency spikes are typically rare. Still, a non-negligible fraction of paths experience substantial latency spikes. The CDF of the inter-quartile ranges (IQRs) of latencies of all timelines reveals that for about 70% of the timelines (Fig. 3c), variations in latencies are marginal (i.e., IQR is only a few milliseconds). Approximately half the paths virtually experienced *no* latency variations. About 5% of the timelines, however, exhibit substantial latency variations—an IQR of at least 100 ms. When we compare IQRs of latencies of inter-ASN paths with those of intra-ASN paths, we observe that the maximum latency variation observed in the intra-ASN paths is at least a factor of four smaller than that in the inter-ASN paths.

Takeaways. *End-to-end latencies of paths between end users and CDN edge servers are typically quite stable—spikes are rare, and variations are small, if any. Nevertheless, a non-negligible fraction of observations indicates significant latency spikes and variations.*

B. Mathematical constancy of latencies

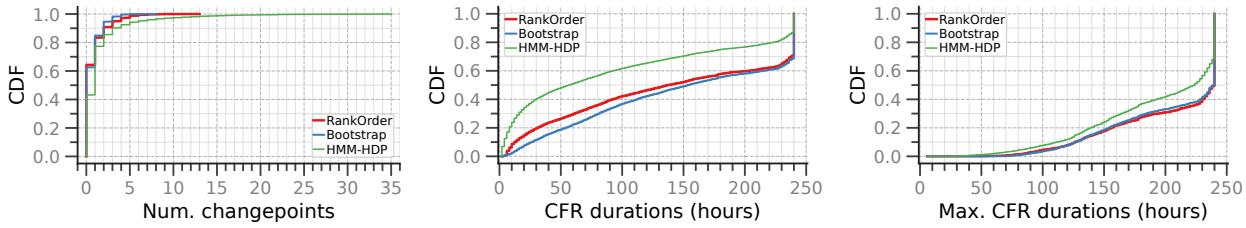
We use three different changepoint detection algorithms—Bootstrap, RankOrder, and HMM-HDP (§III)—for determining where the observed latencies on any given timeline exhibit a *level shift*, if any. We utilized these changepoints to quantify the duration for which the observed RTTs in each timeline stays *constant*. Lastly, the changepoints are performance agnostic: They simply indicate a level shift in end-to-end RTTs, regardless of whether that change increased or decreased the path latency.

Analyzing changepoint detection algorithms. We ran all three algorithms on each timeline in our data set. Per Fig. 2 except for a small fraction of timelines with no latency measurements, timelines are typically complete, with a few (i.e., 358 or 6% of) timelines having one or more missing observations. Most of these timelines also exhibit little, if any, latency variations (Fig. 3c). As a consequence, all three changepoint detection algorithms report *no* level shifts (or changepoints) for about half of the timelines (Fig. 4a). While Bootstrap and RankOrder do not detect any level shifts for nearly 66% of the timelines, the corresponding fraction when using the HMM-HDP method is only 43% of the timelines. HMM-HDP typically detects more changes than the other two methods: The 99th percentile of the number of changepoints detected are 4, 6, and 16.61 for Bootstrap, RankOrder, and HMM-HDP, respectively. These findings concerning the coverage align with observations in prior work [22]. We skipped validation of the accuracy of the HMM-HDP method since prior work has already demonstrated that HMM-HDP fares better than the other two approaches except for a few inaccuracies in the detected changepoints.

Change-free regions (CFRs). We now examine the distribution of the time durations between any two changepoints detected by the three methods (Fig. 4b). Since HMM-HDP detects more changepoints than the other two, the CFR durations reported by this method are smaller than those reported by the other two methods. The median CFR durations for the HMM-HDP method is about 50 hours, while those for the other two are about a factor of three larger (i.e., 140 and 150 hours for the RankOrder and Bootstrap methods, respectively). Compared to the median CFR duration of about 30 minutes (computed using the HMM-HDP method) for latencies of network paths between arbitrary anchors in Davisson et al.'s study [22], the median CFR duration observed across the paths between anchors and CDN edge server are an order of magnitude larger. In other words, on the paths over which end users typically fetch their data over the Internet, the latencies are constant (or stable) for much longer time periods than those of paths between arbitrary vantage points in the Internet. If we combine this observation with those concerning latency spikes and variations, it appears that these end-to-end path latencies are quite stable, but when they change they may experience a significant shift in value. Perhaps these occasional latency spikes and/or variations might be due to inevitable path-level changes in the Internet between the endpoints [27].

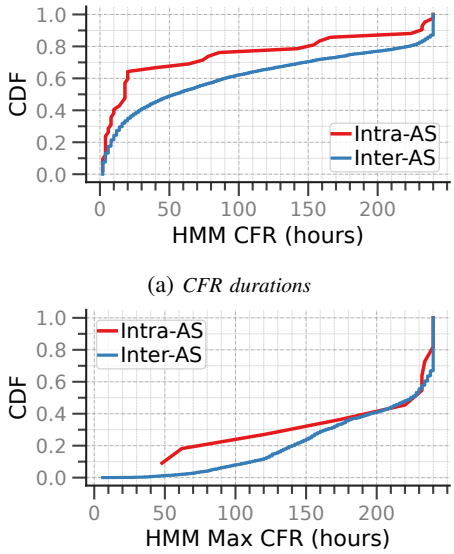
Maximum CFR durations. For each timeline, we then calculated the maximum CFR duration; the larger this value, the longer the time period an endpoint pair experienced near constant latency. The CDFs of maximum CFR durations of all timelines, shown in Fig. 4c, reveal a maximum CFR duration of about 9 days, which is three times larger than the value of 3 days reported in prior work [22]. This observation, especially given that we observe two orders of magnitude larger spikes than those in prior work, may seem contradictory. We note, however, that latency spikes are *transient* "blips" in latency observations, whereas changepoints represent "level shifts" in (baseline) latency observations. Changepoint detection algorithms are designed, by construction, to ignore transient variations; they identify locations in a timeline where latency observations denote a significant and permanent shift in the baseline value. Lastly, since our measurements are two hours apart instead of five minutes, as in prior work, it is quite feasible for us to detect more latency spikes but longer change-free regions than those in prior work.

CFR of inter and intra-AS paths. To analyze the difference in CFR durations, if any, of inter-AS and intra-AS paths, we computed the CDFs of the CFR durations for each of these two categories separately. For this analysis, we restrict



(a) *Number of change points* (b) *CFR durations* (c) *Maximum CFR duration*

Fig. 4: *(a) The HMM-HDP changepoint detection algorithm detects more changes (or level shifts) in the latencies of timelines than either the RankOrder or Bootstrap algorithm; consequently, (b) CFR durations reported by the HMM-HDP algorithm are typically smaller than those reported by the other two methods; and (c) Max. CFR durations reported by the HMM-HDP algorithm are typically smaller than those reported by the other two methods.*



(a) *CFR durations*

(b) *Max. CFR durations*

Fig. 5: *(a) CFR and (b) maximum CFR durations detected by the HMM-HDP methods on inter-AS and intra-AS paths.*

our attention to using just the HMM-HDP method, which is typically more accurate than the others. Per Fig. 5, the intra-AS paths in our measurements reveal a smaller CFR duration than inter-AS paths; the median CFR duration for the former is about 20 hours, a factor of 2.5 smaller than that of the latter, which is about 50 hours. Nearly 40% of the intra-AS paths report a maximum CFR period that is smaller than those of inter-AS paths. Our earlier observations concerning latency spikes and variations (refer Fig. 3) showed that intra-AS paths experienced far fewer latency spikes and variations. These observations perhaps collectively suggest that the intra-AS paths might be experiencing changes more frequently than inter-AS paths; the former are within the control of an AS, and the network operator might routinely perform traffic engineering for optimizing the performance of paths. Though the HMM-HDP methods detects these level shifts in RTTs, the impact on latencies seems rather positive.

Takeaways. *The end-to-end latencies of network paths, over which end users typically consume bandwidth, are quite stable, i.e., free from frequent level shifts in latencies. Compared to prior work [22], the CFR durations of the paths we analyzed are an order of magnitude larger, and the median maximum CFR durations are at least three times larger. The median and maximum CFR durations of Intra-AS paths are smaller than those of inter-AS paths, but, based on §IV-A, these changes do not seem to affect the path RTTs adversely.*

TABLE II: *Max. CFR durations (in hours) under different thresholds of operational constancy; larger the threshold the less "stricter" the notion of constancy*

Thresh.	*avg.*	P_{25}	P_{50}	P_{75}	P_{99}	*sdev.*
100	207.8	228.0	240.0	240.0	240.0	65.6
50	191.1	156.0	240.0	240.0	240.0	77.8
25	173.2	108.0	234.0	240.0	240.0	86.5

C. Operational constancy of latencies

We can consider latencies as *operationally constant* if the variations are within certain bounds (or thresholds) that can be treated as "constant" or "equivalent" from a (network) operational perspective. In Zhang et al.'s work, for instance, they consider latency variations within various intervals such as 0-100 ms and 100-200 ms, etc., and re-evaluate how ignoring the variations within the intervals affect the CFR durations [21]. Davisson et al. repeated this analysis and observed that 95% of their traces had maximum CFR durations of 24 hours or longer. To put our findings in perspective with those from these two prior works, we repeated this analysis under different thresholds or latency intervals that can be deemed operationally equivalent (Tab. II). In performing these analyses, we assume that any two latency observations of a timeline are spaced exactly 2 hours apart. In reality, sometimes the measurements might, however, be spaced closer: The RIPE Atlas platform, for instance, randomizes the starting times and ordering of measurements. The interval between observations should, nevertheless, be about 2 hours on average.

Even under the more stringent notion of mathematical constancy (Fig. 4c), we observe that in the median, the maximum CFR for the HMM-HDP method is about 220 hours (or 9 days). Compared to prior work, per this plot, 95% of the timelines (or paths) report a maximum CFR duration of about

TABLE III: *Prediction error for EWMA with different values for smoothing factor* α

α	$avg.$	P_{50}	P_{99}	$sdev.$
0.5	0.27	0.04	3.76	0.68
0.25	0.38	0.07	4.19	0.80
0.125	0.43	0.09	4.35	0.85
0.01	0.50	0.11	4.39	0.91

100 hours or longer. If we, however, consider latency variations within an interval of 100 ms to be operationally equivalent, the median CFR durations cover the entire measurement period. At this threshold, the timelines basically are devoid of any latency changes or level shifts from an operational perspective. Even if we lower the threshold for operational equivalency to 25 ms, the median maximum CFR duration is only 6 hours short of spanning the entire measurement period.

Takeaways. *End-to-end latencies along paths between end users and CDN edge servers show little to no changes (or level shifts) from an operational perspective. Compared to prior work,* 95% *of the paths have a maximum CFR duration of* 100 *hours or longer—a factor of* 4 *or higher than that reported in prior work.*

D. Predictive constancy of latencies

The predictive notion of constancy measures the error in predicting end-to-end latencies (given a sequence of prior observations). Similar to prior work [22], we use two families of predictors: SMA and EWMA. We then compute the prediction error as follows.

$$\text{Prediction Error} = \mathbb{E}\left[\left|\log\left(\frac{\text{predicted}}{\text{actual}}\right)\right|\right]$$

The choice of the log error comes from prior work [21], [22]. Mean Squared Error (MSE) and Mean Absolute Error (MAE) are absolute metrics of error, while log error is relative. In scenarios where data spans multiple orders of magnitude (e.g., latency measurements), log error is a more interpretable error than absolute metrics such as mean squared error or mean absolute error. We used EWMA with a range of different values for the smoothing factor α, and the prediction errors, regardless of the choice of the smoothing factor, are typically quite small (Tab. III). Similarly, we computed the prediction errors using SMA with different values of window sizes (W), and the errors do not vary much (Tab. IV). The prediction errors for the SMA predictor are, however, larger than those for the EWMA predictor, indicating the simple moving averages do not suffice; small deviations within a concerned window of observations might increase the prediction error.

The measurement intervals in our study are coarse-grained (III-C) compared to prior work [22]—every two hours instead of four minutes. This difference has implications specifically for the accuracy of the analyses concerning the predictive constancy of latency measurements: The prediction errors we observe are larger than those reported by Davisson et al. [22]. We observe that if we use a more involved filter, e.g., the Kalman filter, we can reduce the prediction error despite the coarse granularity of our measurements; as P increases, the predicted values shift closer in magnitude to the actual observations (Tab. V).

TABLE IV: *Prediction error for SMA with window size* W *(in number of data-points)*

W	$avg.$	P_{50}	P_{99}	$sdev.$
2	0.43	0.07	4.93	0.96
4	0.45	0.08	4.68	0.93
8	0.47	0.10	4.57	0.92
16	0.48	0.10	4.55	0.91
32	0.51	0.12	4.55	0.92

TABLE V: *Prediction error with a Kalman filter with process variance (P) and measurement variance (Q)*

P	***Q***	$avg.$	P_{50}	P_{99}	$sdev.$
1e-6	1e-2	0.50	0.12	4.50	0.91
1e-5	1e-2	0.49	0.11	4.48	0.89
1e-4	1e-2	0.44	0.09	4.39	0.86
1e-3	1e-2	0.36	0.07	4.16	0.79
1e-2	1e-2	0.22	0.03	3.47	0.61

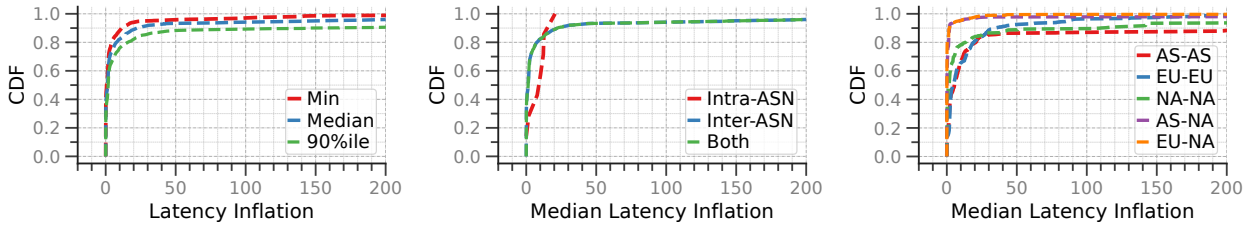
Takeaways. *Our analysis indicates that the error in predicting latency based on the observed latencies is, in the median, quite small. The prediction errors we observe are, however, larger than those reported by Davisson et al. [22], but the differences are likely from the differences in the scale and scope of the studies.*

E. Latency inflation

Until now, we compared the end-to-end RTTs of each timeline to the best of what we observed to be empirically feasible in that timeline. In other words, we characterized how observed latencies deviate from each timeline's minimum empirically measured values. We now compare our latency observations to the *theoretical* minimal latency for characterizing latency *inflations*.

We define latency *latency inflation* as the ratio of the actual observed RTT to the theoretically achievable minimum RTT. For computing the theoretical minimum latency, we assume that the endpoints are connected over the shortest distance on the surface of the Earth (i.e., great-circle distance) using fiber optic cables. The maximum propagation speed of light in fiber optics, given the typical refractive index of the medium, is two-thirds the speed of light in free space. To calculate the distance between endpoints, we must know their physical locations on the surface of the Earth. For the vantage points, we use the user-provided location information on RIPE Atlas. Although RIPE Atlas might intentionally introduce some uncertainty in these location data for privacy reasons, the accuracy of these estimates suffices for our analyses. We gelocated the target IP addresses (i.e., CDN's edge servers) using the IPInfo geolocation service [38]. Given the latitude and longitude coordinates of a pair of endpoints, we determine the great-circle distance between them using the Haversine formula [42].

In principle, the latency inflation (ratio) should be greater than or equal to 1 since the observations will otherwise violate



(a) *Min., med., and 90th perc.* (b) *Med. inflation by path type* (c) *Med. inflation by geography*

Fig. 6: *(a) The CDFs of the minimum, median, and 90th of latency inflations between various endpoints show that for a small, non-trivial fraction of measurements observed latencies are inflated by a factor of 50 or larger; (b) median inflations in RTTs of intra-AS paths are quite smaller than those of inter-AS paths; and (c) median inflations in RTTs of intracontinental paths are higher than those observed on intercontinental paths.*

the fundamental speed-of-light constraint. The CDFs of the minimum, median, and 90th of latency inflations in our study (Fig. 6a), however, reveal that approximately more than half of the latency measurements exhibit an inflation ratio that is lesser than 1. We examined the distances between the endpoints with measurements showing an inflation of less than one and found that they are at least separated by $100\,\mathrm{km}$ and by approximately $9000\,\mathrm{km}$ in the median. The less-than-one inflation ratios, therefore, likely stem from inaccuracies in geolocating the IP addresses of the targets. Such inaccuracies significantly affect the theoretical minimum calculation, leading to unreliable inflation values. These findings highlight the limitations of using geolocation databases for precise latency modeling and suggest that improvements in geolocation accuracy are essential for more reliable latency-inflation analyses.

A small but non-trivial fraction of latency observations, nevertheless, exhibit substantial inflation. The median latency inflations observed for about 10% of the timelines are at least two orders of magnitude larger than what is theoretically feasible (Fig. 6a). Furthermore, the median latency inflation in inter-AS paths is higher than that observed in intra-AS paths (Fig. 6b); the worst-case latencies in inter-AS paths are at least an order of magnitude larger than those observed in intra-AS paths. We then categorized the timelines based on the geographical locations of the vantage points and targets into intercontinental and intracontinental paths. The distribution of median inflations in each of these groups (a subset of which are shown in Fig. 6c) indicates that intercontinental paths experience lesser latency inflations, in the tail, than the intracontinental paths.

Takeaways. *A small but non-trivial fraction of observations exhibit substantial inflation (i.e., more than two orders of magnitude) in latencies. Many of the computed latency inflations are less than one, which also unambiguously suggests the accuracy limitations of freely available geolocation databases.*

V. Conclusion

In this work, we analyzed the end-to-end latencies of the network paths between end users and CDN edge servers. Since CDNs serve most of the content to end users today, a characterization of the latency of these paths will help in estimating the performance of end-user applications and identifying opportunities for improvements. We find that latency spikes are rare, and paths typically experience minimal variations. A small, but non-trivial fraction of endpoint pairs, however, manifests atypical results. About 7.5% of the observations, for instance, deviate by at least a factor of 2 or higher than the median latency observed for the concerned path. Nearly 4% of the observations demonstrate spikes that are a factor of 10 or higher, compared to the median of corresponding paths, in magnitude. Though these findings are quite large compared to those reported (0.01%) in prior work [22], the time periods during which the end-to-end paths in our study exhibited "constant" latencies were at least an order of magnitude larger than those of that prior work. These findings suggest that while CDNs along with infrastructural improvements and protocol optimizations have substantially helped in improving the stability or constancy of latency, when latencies vary (perhaps because of path changes, outages, etc.) the shifts in latencies are sometimes quite substantial.

We hope our work inspires further investigation into the underlying factors contributing to latency or their variations. Our analysis indicated cyclic level shifts in some timelines, with patterns often following daily or weekly periods. These shifts may result from routing changes, such as per-packet load balancing, but could also be due to routine network maintenance activities. To better understand these phenomena, fine-grained experiments on timelines with such level shifts could help distinguish between isolated anomalies and recurring patterns. Replicating this study for IPv6 and also augmenting it with fine-grained path-level analyses can shed light on the factors that affect the end-to-end latencies.

Acknowledgments

We thank the anonymous reviewers and the shepherd for their insightful comments. We also thank Debopam Bhattacherjee for his feedback. We acknowledge the Computer Science and Information Sciences department at BITS Pilani, Goa, and the BITS BioCyTiH Foundation for their support.

References

[1] J. Brutlag, "Speed Matters for Google Web Search," http://services.google.com/fh/files/blogs/google_delayexp.pdf, June 2009.

[2] M. Chow, D. Meisner, J. Flinn, D. Peek, and T. F. Wenisch, "The Mystery Machine: End-to-end Performance Analysis of Large-scale Internet Services," in *ACM OSDI*, Oct. 2014.

[3] L. Hsiao, B. Krajancich, P. Levis, G. Wetzstein, and K. Winstein, "Towards Retina-Quality VR Video Streaming: 15ms Could Save You 80% of Your Bandwidth," *ACM SIGCOMM Computer Communications Review (CCR)*, vol. 52, no. 1, mar 2022.

[4] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Why Is the Internet so Slow?!" in *Passive and Active Measurement (PAM)*, 2017.

[5] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs, "The Internet at the Speed of Light," in *SIGCOMM Workshop on Hot Topics in Networking (HotNets)*, 2014.

[6] T. K. Dang, N. Mohan, L. Corneo, A. Zavodovski, J. Ott, and J. Kangasharju, "Cloudy with a Chance of Short RTTs: Analyzing Cloud Connectivity in the Internet," in *ACM Internet Measurement Conference (IMC)*, November 2021.

[7] Z. Lai, W. Liu, Q. Wu, H. Li, J. Xu, and J. Wu, "SpaceRTC: Unleashing the Low-latency Potential of Mega-constellations for Real-Time Communications," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2022.

[8] R. de Oliveira Schmidt, J. Heidemann, and J. H. Kuipers, "Anycast Latency: How Many Sites Are Enough?" in *Passive and Active Measurement (PAM)*, 2017.

[9] M. Xu, Z. Fu, X. Ma, L. Zhang, Y. Li, F. Qian, S. Wang, K. Li, J. Yang, and X. Liu, "From cloud to edge: a first look at public edge platforms," in *ACM Internet Measurement Conference (IMC)*, 2021.

[10] M. Claypool, "The effect of latency on user performance in Real-Time Strategy games," *Computer Networks*, vol. 49, no. 1, 2005.

[11] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu, "The effect of latency on user performance in Warcraft III," in *Proceedings of the 2nd Workshop on Network and System Support for Games*, 2003.

[12] X. Bai, I. Arapakis, B. B. Cambazoglu, and A. Freire, "Understanding and Leveraging the Impact of Response Latency on User Behaviour in Web Search," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, Aug. 2017.

[13] X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, "Demystifying Page Load Performance with WProf," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013.

[14] T. Flach, N. Dukkipati, A. Terzis, B. Raghavan, N. Cardwell, Y. Cheng, A. Jain, S. Hao, E. Katz-Bassett, and R. Govindan, "Reducing Web Latency: The Virtue of Gentle Aggression," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, August 2013.

[15] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker, "Low Latency via Redundancy," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13, 2013.

[16] D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "cISP: A Speed-of-Light Internet Service Provider," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2022.

[17] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control," *ACM Queue*, vol. 14, no. 5, Oct. 2016.

[18] V. Jacobson, "Congestion avoidance and control," in *Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 1988.

[19] V. Arun, M. Alizadeh, and H. Balakrishnan, "Starvation in End-to-End Congestion Control," in *Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2022.

[20] D. Zeynali, E. N. Weyulu, S. Fathalli, B. Chandrasekaran, and A. Feldmann, "Promises and Potential of BBRv3," in *Passive and Active Measurement (PAM)*, 2024.

[21] Y. Zhang and N. Duffield, "On the Constancy of Internet Path Properties," in *ACM Internet Measurement Workshop (IMW)*, 2001.

[22] L. Davisson, J. Jakovleski, N. Ngo, C. Pham, and J. Sommers, "Reassessing the Constancy of End-to-End Internet Latency," in *Network Traffic Measurement and Analysis Conference (TMA)*, September 2021.

[23] E. Pujol, I. Poese, J. Zerwas, G. Smaragdakis, and A. Feldmann, "Steering hyper-giants' traffic at scale," in *ACM CoNEXT*, ser. CoNEXT '19, 2019.

[24] W. Aqeel, B. Chandrasekaran, A. Feldmann, and B. M. Maggs, "On Landing and Internal Web Pages: The Strange Case of Jekyll and Hyde in Web Performance Measurement," in *ACM Internet Measurement Conference (IMC)*, 2020.

[25] A. Bhat, V. Ganatra, A. Shaha, B. Chandrasekaran, and V. Naik, "On the Constancy of Latency at the Internet's Edge: Tools and Analyses," https://github.com/adityabhat3/const-lat, 2025.

[26] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, Oct 1997.

[27] B. Chandrasekaran, G. Smaragdakis, A. Berger, M. Luckie, and K.-C. Ng, "A Server-to-Server View of the Internet," in *ACM CoNEXT*, December 2015.

[28] M. Appel, E. Aben, and R. Fontugne, "Metis: Better atlas vantage point selection for everyone." in *Network Traffic Measurement and Analysis Conference (TMA)*, 2022.

[29] N. Martin and F. Dogar, "Divided at the edge-measuring performance and the digital divide of cloud edge data centers," *Proceedings of the ACM on Networking*, vol. 1, no. 3, 2023.

[30] R. Ingabire, A. Bazco-Nogueras, V. Mancuso, L. M. Contreras, and J. Folgueira, "Clearing clouds from the horizon: Latency characterization of public cloud service platforms," in *IEEE International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024.

[31] R. Bose, S. Fadaei, N. Mohan, M. Kassem, N. Sastry, and J. Ott, "It'sa bird? it'sa plane? it's cdn!" *SIGCOMM Workshop on Hot Topics in Networking (HotNets)*, 2024.

[32] RIPE NCC, "RIPE Atlas Docs," https://atlas.ripe.net/docs/apis/rest-api-manual/README, May 2024.

[33] ——, "Host a Probe," https://www.ripe.net/analyse/internet-measurements/ripe-atlas/host-a-probe/, 2024, [Last accessed on November 20, 2024].

[34] T. Holterbach, C. Pelsser, R. Bush, and L. Vanbever, "Quantifying Interference Between Measurements on the RIPE Atlas Platform," in *ACM Internet Measurement Conference (IMC)*, 2015.

[35] HTTP Archive, "CDN — The Web Almanac," https://almanac.httparchive.org/en/2022/cdn, Oct. 2022, [Last accessed on November 20, 2024].

[36] Selenium, "Selenium WebDriver," https://www.selenium.dev/, 2025, [Last accessed on March 1, 2025].

[37] MaxMind, "Industry leading IP Geolocation and Fraud Prevention — MaxMind," https://www.maxmind.com/en/home, 2025, [Last accessed on March 1, 2025].

[38] IPInfo, "Trusted IP Data Provider — IPInfo," https://ipinfo.io/, 2025, [Last accessed on March 1, 2025].

[39] HTTP Archive, "Starting your own Measurements (User-defined Measurements) — Quotas," https://atlas.ripe.net/docs/getting-started/user-defined-measurements.html#viewing-a-measurement, Jun. 2024.

[40] M. Mouchet, S. Vaton, T. Chonavel, E. Aben, and J. Den Hertog, "Large-scale characterization and segmentation of internet path delays with infinite hmms," *IEEE Access*, vol. 8, pp. 16 771–16 784, 2020.

[41] M. Mouchet, "Infinite Hidden Markov Models for Julia," https://github.com/SmartMonitoringSchemes/HDPHMM.jl, Jun. 2023.

[42] Wikipedia, "Haversine Formula," https://en.wikipedia.org/wiki/Haversine_formula, 2024, [Last accessed on March 1, 2025].