A Server-to-Server View of the Internet

Balakrishnan Chandrasekaran
Duke University
balac@cs.duke.edu

Georgios Smaragdakis
MIT / TU Berlin / Akamai
gsmaragd@csail.mit.edu

Arthur Berger
MIT / Akamai
awberger@csail.mit.edu

Matthew Luckie
University of Waikato
mjl@wand.net.nz

Keung-Chi Ng
Akamai
kng@akamai.com

ABSTRACT

While the performance characteristics of access networks and end-user-to-server paths are well-studied, measuring the performance of the Internet's *core* remains, largely, an uncharted territory. With more content being moved closer to the end-user, server-to-server paths have increased in length and have a significant role in dictating the quality of services offered by content and service providers. In this paper, we present a large-scale study of the effects of *routing* changes and *congestion* on the end-to-end latencies of server-to-server paths in the core of the Internet.

We exploit the distributed platform of a large content delivery network, composed of thousands of servers around the globe, to assess the performance characteristics of the Internet's core. We conduct measurement campaigns between thousands of server pairs, in both forward and reverse directions, and analyze the performance characteristics of server-to-server paths over both long durations (months) and short durations (hours). Our analyses show that there is a large variation in the frequency of routing changes. While routing changes typically have marginal or no impact on the end-to-end round-trip times (RTTs), 20% of them impact IPv4 (IPv6) paths by at least $26\,\mathrm{ms}$ ($31\,\mathrm{ms}$). We highlight how dual-stack servers can be utilized to reduce server-to-server latencies by up to $50\,\mathrm{ms}$. Our results indicate that significant daily oscillations in end-to-end RTTs of server-to-server paths is not the *norm*, but does occur, and, in most cases, contributes about a $20\,\mathrm{ms}$ increase in server-to-server path latencies.

CoNEXT '15 December 01-04, 2015, Heidelberg, Germany

DOI: 10.1145/2716281.2836125

CCS Concepts

•**Networks** → **Network measurement;** *Network performance analysis;* Network dynamics;

Keywords

Internet's core; routing changes; server-to-server paths; dual-stack; congestion

1. INTRODUCTION

The Internet is massively heterogeneous and also continuously evolving, and thus, no one vantage point can capture the breadth of these changes [7]. Many recent studies utilize vantage points located at end users to launch measurement campaigns. These studies provide an end-user's perspective on performance of broadband Internet [43], ISP bandwidth cap and throttling [30], download time for popular content primarily delivered by content delivery networks (CDNs) [44, 37], offloading of services and computation to the cloud [41, 20], and video streaming quality [4, 3, 19]. Tracking changes in the Internet's core and assessing their impact on network performance and operation, however, is not as well studied.

A major obstacle to studying the state of the Internet's core is the limited set of vantage points that can be utilized to support such a study. Looking glass servers [24, 9], many of which are located at core routers, can offer visibility into the core. They are, however, designed for testing basic reachability and not measure end-to-end path performance. Moreover, they are typically not suitable for frequent and large-scale experiments. It is also challenging to collect end-to-end measurements in both forward and reverse directions between two end points. Although a number of measurement platforms, viz., PlanetLab, are available, their network coverage is limited as most of the measurement servers on these platforms are installed in residential and academic networks. Other distributed platforms viz., RIPE Atlas, are widely deployed and hence, provide better coverage. There are concerns, nevertheless, about the accuracy of delay-based measurements when using shared measurement platforms [21]. Cloud servers are also known to be over-utilized and virtual machines can be transparently migrated to different physical servers [25]; hence, these are not good vantage points for accurate delay-based measurements.

Studying the performance characteristics of router interconnections between networks in the Internet's core at scale would require the installation of thousands of *physical* servers around the globe at a diverse set of peering locations including colocation centers, datacenters, Internet exchange points (IXP), as well as inside eyeball networks. While the idea of installing servers in a number of locations for Internet measurements is not new and has been shown to provide good insights [35], the investment that is required to install servers in a large number of peering locations and networks is prohibitively high.

In this paper, we investigate the state of the Internet's core at scale. To this end, we report on a unique view of the Internet's core by utilizing thousands of servers deployed by a large commercial CDN. We routinely performed server-to-server measurements for more than 16 months, in both forward and reverse direction, and report on the state of the Internet core from a service provider perspective. In particular, we study the affect upon server-to-server round-trip times of (1) routing changes, and (2) significant daily oscillations in latency, herein called congestion. Our work provides a complementary view of the Internet at a scale, where a significant fraction of the Internet traffic flows [38], that is currently difficult to obtain by performing measurements at the edge of the Internet.

The contributions of this paper are as follows:

- We study the effect of routing changes in the core of the Internet on hundreds of thousands of server-to-server paths over both long and short time scales. In our data, the performance degradation due to routing changes is typically low. 4% (7%) of routing changes on IPv4 (IPv6), however, increase RTTs by at least $50\,\mathrm{ms}$ for at least 20% of the study period.
- We use delay-based methods described in [17, 27] to analyze hundreds of thousands of server-to-server pairs for congestion events, use a preliminary router ownership technique to infer the ASes operating the routers involved, and characterize the links based on the ASes and relationships inferred. In our data, congestion is not the *norm* in the Internet's core, but when it occurs, we detect it in the interior of networks as well as on the interconnection between two networks. In the context of the latter, the congestion occurs more often on private peering links. Congestion, in most cases, contributes about a $20\,\mathrm{ms}$ increase in server-to-server path latencies.
- Complementary to other studies such as [16], we find that the overall server-to-server path performance over both IPv4 and IPv6 protocols is converging. We also highlight opportunities to reduce server-to-server path latencies by up to 50 *ms*, using dual-stacked servers.

Although the results presented in this paper are based on a large-scale study that involves thousands of vantage points located in diverse networks, as close as possible to the core, we do not argue that the data sets offer a representative view of the Internet's core. We hope that this study will inspire other follow-up efforts, each presenting different views of the complex role of the Internet's core, that can be complementary to this work.

2. DATA SETS

The data sets in this study were obtained from servers of the measurement platform of a large commercial CDN. The CDN operates servers in more than 2000 diverse locations including colocation centers, Internet exchange points (IXP), datacenters and hosting facilities. At each location there may be one or more *server clusters*. Most of the servers are dual-stack systems, supporting both IPv4 and IPv6. The CDN operates approximately $10K$ server clusters and $150K$ servers around the globe.

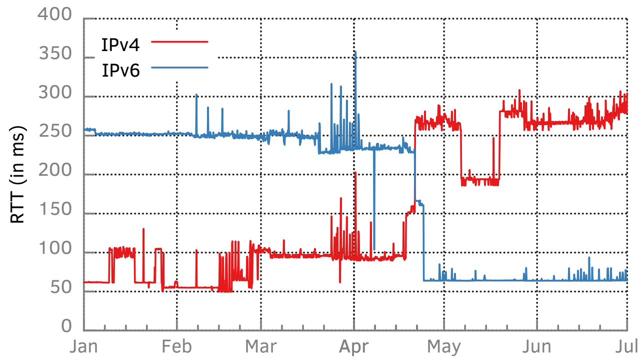
For operational reasons, one server at each cluster is utilized to perform measurements (traceroutes and pings) to DNS servers and other CDN servers. These measurements serve as input to the CDN's *mapping* system, which is responsible for determining how to map end-user requests to appropriate CDN servers [34, 14]. We used these measurements and supplemented them with customized traceroute campaigns conducted from the same measurement servers.

2.1 Long-term Data Set

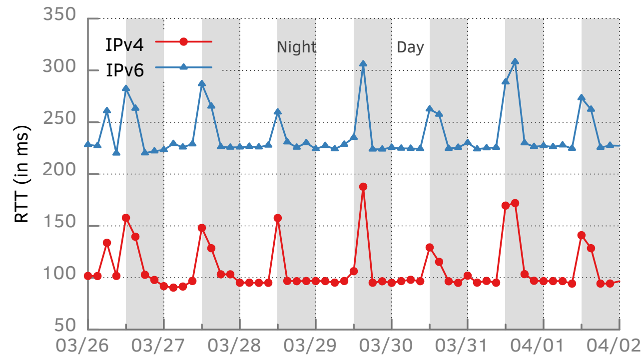
To capture long-term performance characteristics of server-to-server paths we used traceroutes gathered between all pairs (*full mesh*) of approximately 600 dual-stack CDN servers. The servers, each located in a different server cluster, were selected from over 70 different countries with approximately 39% of the servers located in the USA. Australia, Germany, India, Japan and Canada are the next top five countries, in order, by the number of servers present in each country and taken together they represent 19% of the total number of servers used in the measurement study.

The traceroutes were scheduled once every three hours between all pairs of servers over both IPv4 and IPv6 for 16 months, from January 2014 through April 2015. All traceroutes performed during a collection period are grouped together and annotated with an identical timestamp. The data set contains approximately $2.6B$ traceroutes. A wide range of factors, e. g., hardware and software maintenance activities, and network connectivity issues, affect any such long-term and large-scale data collection effort and reduce the volume of data gathered as well as the percentage of traceroutes that are completed. In this study we considered only the nearly $2B$ (75%) traceroute measurements that are *complete* (the traceroutes reach the intended destinations).

From the router interfaces observed in traceroute, we inferred the autonomous system (AS) path by mapping the IP addresses at each hop to an AS number (ASN) corresponding to the origin AS of the longest matching prefix observed in BGP for each IP address. Possible errors introduced by our simple AS path inference [32, 45, 22, 11] may have no impact with respect to our detection of changes in AS path;



(a) *A six month timeline of RTTs between servers exhibiting level shifts (compare RTTs in March with that in June).*



(b) *A small section of the RTTs timeline exhibiting daily variations (observe the repeating mid-day increases).*

Figure 1: *End-to-end RTTs observed in ICMP traceroutes done from a server in Hong Kong, HK to a server in Osaka, JP.*

#traceroutes with	***IPv4***	***IPv6***
complete AS-level data	(70.30%) 741	(64.03%) 596
missing AS-level data	(1.58%) 16	(3.32%) 30
missing IP-level data	(28.12%) 296	(32.65%) 304

Table 1: *Summary of traceroutes (in millions) collected between dual-stack servers*

though some errors can cause either (1) detecting AS path changes that did not occur or (2) missing changes that did occur. In case of the former, when we compare the measured RTTs for the two, inferred (but really the same) AS paths, we would tend to infer only a small, if any, change in the RTTs. Thus, this tends to increase our inferred proportion of AS-path changes that had little impact on RTT's, and thus under-represent the proportion that had significant impact. As instances of significant impact are of greater interest to us, we would just as soon have the bias in this direction. If the error is when we do not detect a change in AS path, we would erroneously lump together RTTs that were from different AS paths. This tends to inflate the estimate of the variability of RTTs and of the duration of the AS path.

Table 1 presents summary statistics of the data set and shows that the majority of traceroutes, 70% over IPv4 and 64% over IPv6, had complete AS-level data – these traceroutes contained no unresponsive hops, and all IP addresses were covered by a prefix in BGP. A small fraction of traceroutes (row labeled *missing AS-level data*) contained addresses with no known IP-to-ASN mapping. A significant portion of traceroutes, 28.12% over IPv4 and 32.65% over IPv6, contained unresponsive hops. Traceroutes with data missing at AS-level (because of no known IP-to-ASN mapping) or IP-level (unresponsive hops), however, can still be used, and we discuss how we handle these traceroutes in Section 4.

We used *classic traceroute*, except starting in November 2014 we used *Paris traceroute* [8] for IPv4. Routers performing per-flow load balancing on a path between two servers can cause classic traceroute to report erroneous IP-level paths [8]. The AS path inferred from classic traceroute data can contain loops in AS paths, though it is rare for the classic traceroute algorithm to be the cause [28]. A small fraction of traceroutes, 2.16% over IPv4 and 5.5% over IPv6, contain AS-path loops and were not included in the analyses. Because our measurement platform is a production CDN, we can neither run experiments with modified versions of supported protocols (or tools), nor add support for other protocols (or install new tools), viz., *Tokyo Ping* [36].

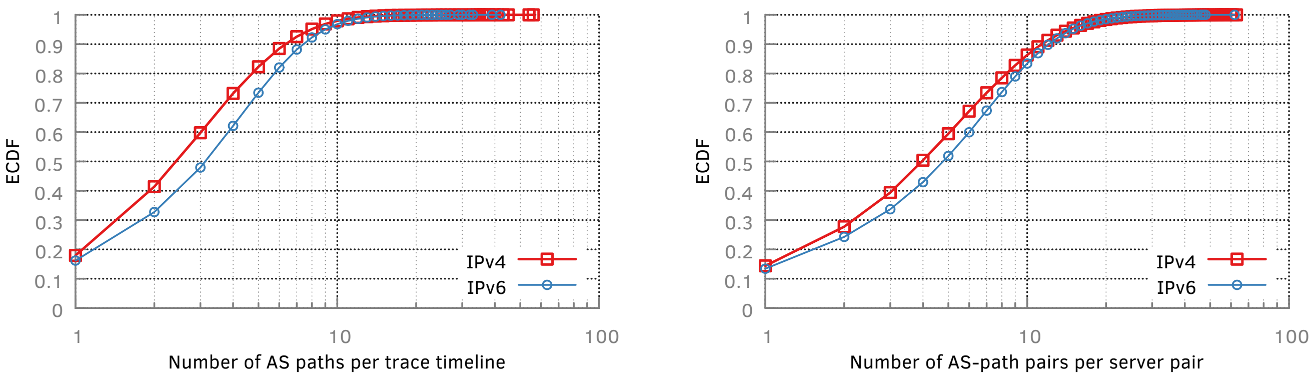
2.2 Short-term Data Set

We performed a series of measurements over smaller time scales (one or more weeks) to measure short-term trends in performance characteristics of server-to-server paths. We analyzed server-to-server ping data collected by the CDN from February 22^{nd} through 28^{th}, 2015. Servers from each one of the several thousand clusters around the world *ping* a predetermined set of servers in other clusters every 15 minutes to gather performance statistics. The ping data set contained more than $2.9M$ IPv4 and approximately $1M$ IPv6 server pairs, each of which had at least 600 measurements (i. e., nearly 90% or more of the total 672 possible measurements per server pair).

Using the ping measurements, we identified $100K$ server pairs where we observed diurnal patterns in the end-to-end RTTs, indicative of congestion (see Section 5). We chose a subset of $50K$ server pairs to ensure that the traceroute measurements between the selected pairs complete in under 30 minutes. The selection consists of servers from around $3.5K$ server clusters, located in more than 1000 locations and 100 countries. We repeated the traceroutes, over both IPv4 and IPv6, between the selected server pairs, in either direction, once every 30 minutes for more than two consecutive weeks. Finally, to infer congestion between clusters at the same location we performed traceroute campaigns between all servers (full mesh) colocated at the same datacenter or peering facility with a frequency of 30 minutes for a period of 20 days.

3. AN ILLUSTRATIVE EXAMPLE

To illustrate the complex performance characteristics of server-to-server paths in the core of the Internet, consider the example in Figure 1a. Traceroutes were performed every three hours, in each direction, over IPv4 and over IPv6,



(a) *ECDF of AS paths per trace timeline.*

(b) *ECDF of AS-path pairs per server pair.*

Figure 2: *Number of unique AS paths and AS-path pairs observed in the* 16*-month period studied.* 80% *of trace timelines have* 5 *or fewer AS paths in IPv4, and* 6 *or fewer in IPv6. Pairing AS paths in the forward and reverse directions still reveals* 80% *of server pairs to have* 8 *or fewer path pairs in IPv4, and* 9 *or fewer in IPv6.*

between dual-stack servers, one located in a datacenter in Hong Kong and the other in Osaka, Japan. The figure shows the RTTs between the endpoints (from Hong Kong to Japan) for the first six months of 2014. Focusing first on the RTTs over IPv4 (the red line), an obvious feature is *level shifts* between periods of a *baseline* RTT with variability above the baseline. During periods where the baseline RTT was above $150\,\mathrm{ms}$ the traceroute went via the west coast of the USA. We inferred the AS paths from the traceroutes, and at each of the level shifts there was a change in the AS path in one, or both, directions. Also, there were cases where the AS path changed, but there was a negligible change in the RTTs. This leads to the first theme of this paper: *to what extent do changes in the AS path affect round-trip times?*

Another key feature of the plot is the *spikes* in RTT, which are a typical feature of repeated measurements. Figure 1b shows the daily oscillation in RTT (as opposed to individual spikes) during the period between March 26 and April 2, 2014; this also occurs from February 14^{th} to 22^{nd}. A daily oscillation in RTT is often an indication of congestion during the busy period of the day somewhere along the path, or at an endpoint[1]. A second theme of this paper is: *how common are periods of daily oscillation in RTT, and where do they occur?*

One can ask the higher level question: *what affects end-to-end performance more - routing or congestion?* In Figure 1, changes in routing seemed to have a greater impact on RTT.

Now consider the RTTs measured over IPv6 (the blue line). To first order, it has similar features as IPv4 (the red line). A notable level shift is on April 21, 2014, where the route for IPv6 got much better at the same time that it got worse for IPv4; RTTs increased by $108\,\mathrm{ms}$ over IPv4, and decreased by $168\,\mathrm{ms}$ over IPv6. During March 26^{th} to April 2^{nd}, the path over IPv6 also experienced a daily oscillation in RTTs, which could have been occurring at equipment that was in both the IPv4 and IPv6 path. A third theme of this paper is: *how does IPv4 and IPv6 compare with respect to routing and performance?*

We would like to stress that the example we selected for illustration serves only as a candidate to highlight interesting observations, the challenges inherent in observing and quantifying them, and interesting research questions that arise. While it is trivial to analyze manually a few cherry-picked examples, it becomes impractical after considering only a few tens of server pairs.

4. IMPACT OF ROUTING CHANGES

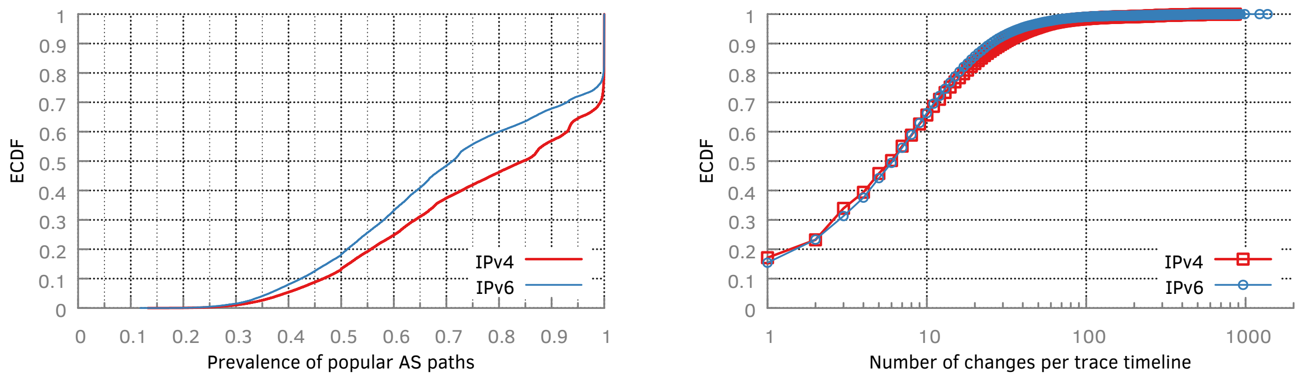
We investigated, using the long-term data set, the impact of routing changes in the core on end-to-end RTTs between server pairs. For simplicity, we restricted our attention to a set of $60K$ server-pairs that, for at least 400 days (of the 485 days of data collection) had, on each day, at least one traceroute between them in both directions and over both IPv4 and IPv6 protocols. We conclude this section with a brief analysis using the short-term data set showing that the coarse granularity of measurements in the long-term data set likely does not affect our results.

4.1 Methodology for Inferring Changes

To capture routing changes along the path between any two servers, we treat the AS paths (with each hop representing a different ASN) between the servers as delimited strings and use the *edit distance* between any two AS paths as a measure of the difference between them. A zero edit distance implies that the AS paths are the same (no change), while a non-zero value implies a different AS-level route.

Suppose AS paths $p_1 : ASN_a \rightarrow ASN_b \rightarrow ASN_c \rightarrow ASN_d$ and $p_2 : ASN_a \rightarrow ASN_b \rightarrow ASN_d$ were observed in traceroutes between two servers A and B at time t_1 and t_2, respectively. The edit distance computation on the path-strings yields the value one, implying that the paths are dif-

[1]The probes were ICMP, and routers may handle them differently from UDP and TCP. Thus, traffic to/from end users may not experience the same degradation as the ICMP probes. Nevertheless, that the ICMP packets did experience the daily oscillation in RTT is evidence of some stress on some equipment on the path, and could foreshadow degradation in performance for the user traffic.



(a) *ECDF of the prevalence of popular AS paths: most paths had one dominant route, with* 80% *dominant for at least half the period.*

(b) *Routing changes in the long-term:* 80% *of the trace timelines experienced* 20 *or fewer changes over the course of* 16*-months.*

Figure 3: *Frequency of routing changes and prevalence of popular AS paths over the course of* 16 *months.*

ferent and one change (removal of ASN_c) is required to make p_1 identical to p_2. We also assume that the routing change from p_1 to p_2 happened at t_2. Traceroutes may contain one or more hops either with no IP address (i. e., non-responsive hop) or with an IP address having no known IP-to-ASN mapping. Although, we cannot eliminate all missing data, we impute the missing hop (at only the AS-level) in instances where either side of the missing hop is the same ASN.

Computing lifetimes. Since our data set contains only one traceroute between any two servers during each three-hour time period, we assumed an AS path observed from a traceroute to persist for three hours. For instance, AS path p_1, in the example above, is assumed to persist during the time interval $[t_1, t_2)$, assuming t_1 and t_2 are consecutive three-hour time intervals (i. e., $t_1 + 3$ *hours* $= t_2$). The *lifetime* of an AS path with respect to a set of endpoints is defined as the total time during which the AS path was observed between the endpoints. For instance, if path p_1 was observed 800 times (the periods during which p_1 was observed do not need be contiguous) in traceroutes from server A to B, the lifetime of p_1 is calculated as 2400 *hours* (100 *days*). The AS-path in the other direction (from B to A) might have a different lifetime depending on how many times, if any, the path was observed. We also refer to the set of all traceroutes from one server to another (representing a *time series*) as a *trace timeline*. As an example, all traceroutes with source as server A and destination as server B over the period of 16-months constitute one trace timeline.

4.2 Data Trends over Long-term

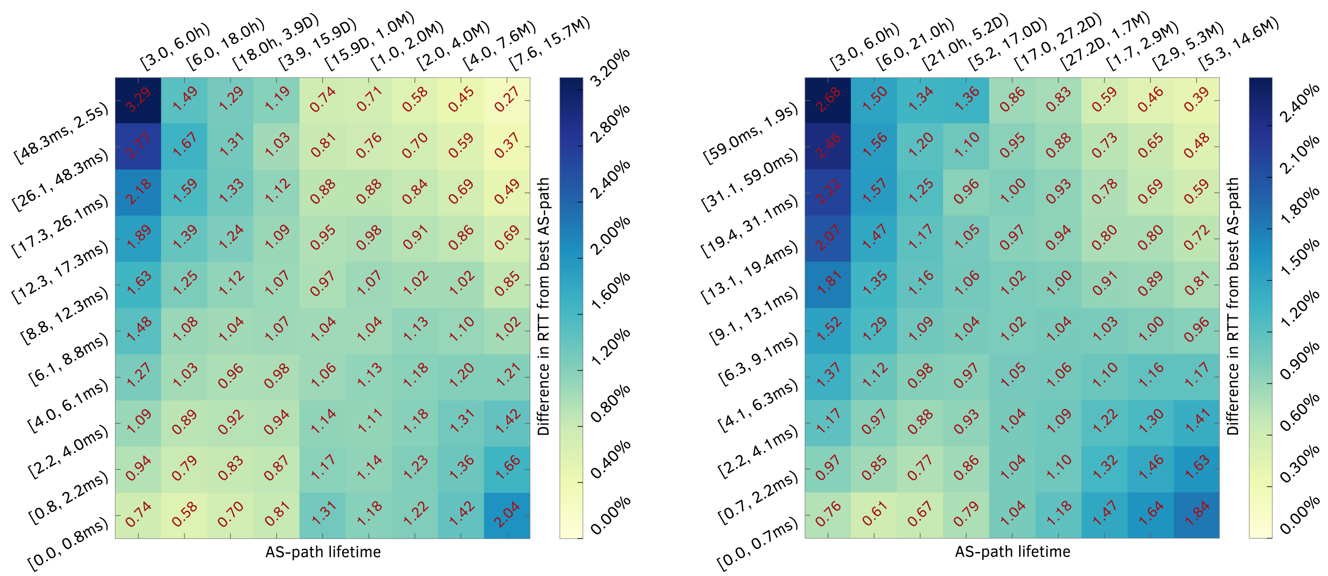
Unique AS paths. For each trace timeline in the long-term data set, we count the unique AS paths and show the ECDF of the path counts in Figure 2a. We observe that 18% (16%) of trace timelines over IPv4 (IPv6) contained only one AS path implying no route change (at AS-level) over the entire duration of the data set. Approximately 80% of the trace timelines have 5 or fewer different AS paths over IPv4 and 6 or fewer over IPv6 over 16 months. Only 2% of the trace-timelines over IPv4 (3% over IPv6) have 10 or more different AS paths.

Since the paths along the forward and reverse directions between two servers can be asymmetric, we associated the AS path observed in the forward direction with that observed at the same time (in a traceroute) in the reverse direction (between the same endpoint pair) and count the number of unique AS-path pairs. From the ECDF of the path pairs, in Figure 2b, we find that 80% of $60K$ server pairs in our data have 8 or fewer different AS-path pairs over IPv4 (and 9 or fewer over IPv6). We observe that routing changes only cause paths between any two servers to fluctuate between a small set of AS paths.

Prevalence of popular AS paths. Similar to [35], we define *prevalence* of an AS path to be the overall likelihood that a particular AS path is observed between a server pair. Figure 3a illustrates the ECDF of the prevalence of the popular AS paths of all trace timelines where *popular* AS path of a trace timeline refers to the AS path with the longest lifetime. The prevalence of the most popular AS paths was at least 50% for 80% of trace timelines–most trace timelines have one dominant route or AS path. In less than 20% of the trace timelines the prevalence of the most popular path was less than 50% over IPv4 and 55% over IPv6; there is a greater likelihood that these trace timelines experienced more routing changes (since the prevalence of the most popular AS paths was less than 50%). Popular paths over IPv6 were observed for relatively shorter durations in comparison to those over IPv4.

Frequency of routing changes. For each trace timeline, we sort the AS paths (of each traceroute) by time and compare the difference (edit distance) between any two AS paths appearing consecutively in time. A non-zero value for difference indicates a change in route between two corresponding servers. The ECDF of the total number of route changes per trace timeline is shown in Figure 3b, and 18% of trace timelines over IPv4 and 16% over IPv6 have no change for the entire time span of 16 months. Nearly 90% of trace timelines have 30 or fewer changes over both protocols–if uniformly distributed in time, that is still less than two changes per month.

Effect of routing changes on RTT. We retrieve the end-to-end RTT and the AS path from each traceroute, and aggregate the RTTs by AS path, separately for each trace time-



(a) *AS-paths over IPv4*

(b) *AS-paths over IPv6*

Figure 4: *Comparing magnitudes of increase in (baseline)* 10^{th} *percentile of RTTs of AS paths (each relative to the best AS path of the corresponding trace timeline) with the lifetime of AS paths. Suffixes 'h', 'M', and 'D' imply 'hours', 'months' and 'days'. Sub-optimal AS paths with significantly higher RTTs (top rows) are often short-lived (left-most columns).*

line. This yields, for each trace timeline, one or more AS path *buckets* each of which is associated with a set of RTTs, and we compute the 10^{th} and 90^{th} percentiles of RTTs in each bucket. The 10^{th} percentile of a bucket captures the *baseline* RTT (refer Section 3) when traversing the particular AS path (associated with the bucket), while the 90^{th} percentile takes into account the *spikes* (refer Section 3) in RTT over the same path. Using the 10^{th} percentile of RTTs as a heuristic, we denote, for each trace timeline, the AS path with the lowest 10^{th} percentile as the *best* AS path (or the *optimal* route). Here "best" is in the context of the paths that were actually observed; we do not consider hypothetical or potential AS paths.

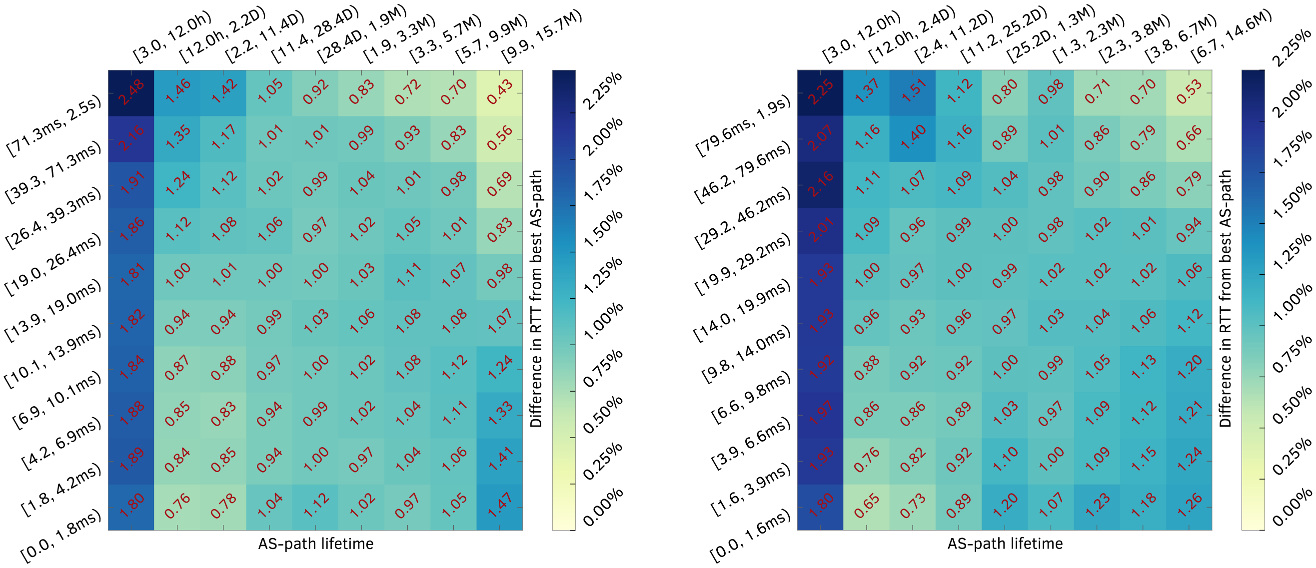
In any given trace timeline, hence, the difference between the 10^{th} percentile of the other AS paths and that of the best AS path, quantifies the increase in RTT incurred as a result of traversing a *sub-optimal* path. Naturally, trace timelines with only one AS path are not included in this analysis. By combining (1) the lifetimes of each sub-optimal AS path for all trace timelines, with (2) the increase in baseline RTT of that sub-optimal path compared to that of the best AS path for the trace timeline, we can analyze the correlation between the two variables. Figure 4 shows, in the form of heat maps, the scatter plot of these two variables for IPv4 and IPv6.

In both heat maps, the X-axis shows bins corresponding to different *deciles* of the distribution of AS-path lifetimes, and the Y-axis shows bins associated with the deciles of the distribution of magnitudes of increase in 10^{th} percentile of RTTs of AS paths (each relative to the best AS path of the corresponding trace timeline). Each bin along the axes represent half-open intervals. The X-axis bin (or interval) $[0.0, 3.0h)$ is not included because it has no data points; the minimum for AS-path lifetime is 3 hours. The $0^{th}\%$ and $10^{th}\%$ of the AS-path lifetime distribution have the same value of 3 hours. Hence, in the heat map, the first column, corresponding to the interval $[3.0, 6.0h)$, represents the first two deciles of the AS-path lifetime distribution.

The value of each cell in the heat map shows the fraction of all AS paths between the server pairs that exhibited an increase in the RTT compared to the best path of the server pair for a given length of time; the Y-axis reports the increase in RTT, and the X axis reports the length of time covered by that increase in RTT. For instance, in Figure 4a, the sixth cell from the left in the second row from the top shows that only 0.76% of all (sub-optimal) AS paths observed at least for one month and at most two months resulted in an increase of at least $26.1\,\mathrm{ms}$ and at most $48.3\,\mathrm{ms}$, when compared to the best AS paths of the corresponding trace timelines.

Figure 4 shows that, for both IPv4 and IPv6 protocols, the baseline RTTs of AS paths with longer lifetimes (in the bottom right corner) are close in value to that of the best AS path of corresponding trace timelines. Paths with poor-performance (large differences in 10^{th} percentiles from the that of the best AS paths are often those with relatively short lifetimes (in the top left corner). Summing the values along a particular row provides the percentage of AS paths with increase in baseline RTTs corresponding to the Y-axis value of that row, and we observe that 10% of the AS paths suffer an increase of at least $48.3\,\mathrm{ms}$ in baseline RTTs over IPv4 and $59\,\mathrm{ms}$ over IPv6. 20% of the paths suffer an increase of at least $25\,\mathrm{ms}$ in baseline RTTs. The heat maps also indicate that both IPv4 and IPv6 exhibit similar patterns–sub-optimal AS paths that result in significant increases in RTT are often short-lived.

Figure 5 similarly shows a heat map for the differences in



(a) *AS-paths over IPv4.*

(b) *AS-paths over IPv6.*

Figure 5: *Comparing magnitudes of increase in* 90^{th} *percentile of RTTs of AS paths (each relative to the best AS path of the corresponding trace timeline) with the lifetime of AS paths. Suffixes 'h', 'M', and 'D' imply 'hours', 'months' and 'days'. As the lifetime of AS paths increases the likelihood of the paths being sub-optimal (not offering the lowest* 90^{th} *percentile of RTT compared to other paths between the corresponding server pair) decreases.*

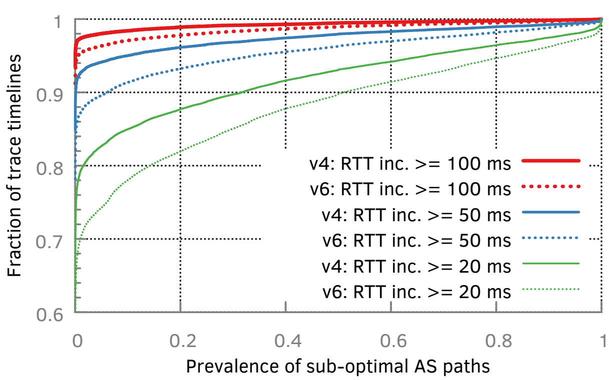


Figure 6: *Sub-optimal AS paths: approximately* 1% *of trace timelines over IPv4 and* 2% *over IPv6 experienced at least a* $100\,\text{ms}$ *increase in RTT because of sub-optimal AS paths with prevalence of* 20% *or more.*

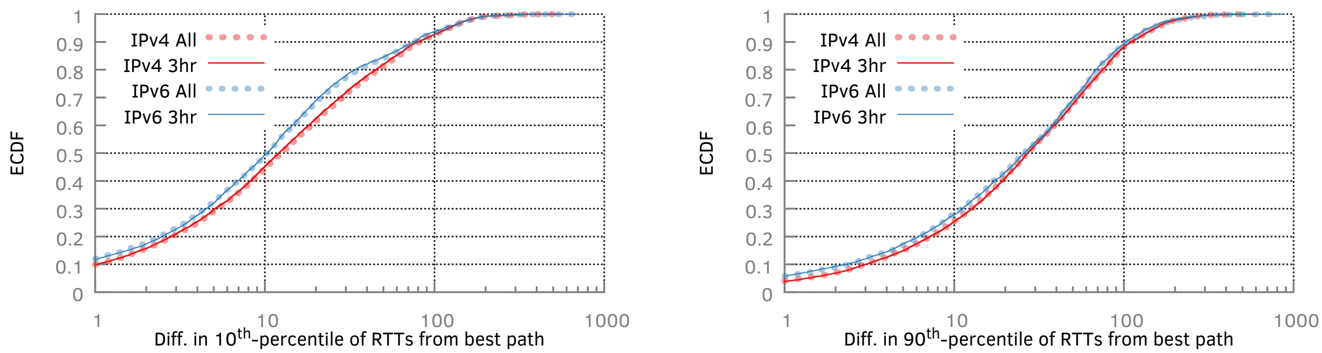
the 90^{th} percentile of RTTs of AS paths relative to the best AS paths (with the lowest 90^{th} percentile value), of corresponding trace timelines. The heat map for the 90^{th} percentile differences exhibits trends comparable to that of the 10^{th} percentile differences. 10% of the AS paths have at least $70\,\text{ms}$ increase in the 90^{th} percentiles of RTTs compared to that of the best paths of corresponding trace timelines. Instead of 10^{th} and 90^{th} percentiles, if we choose standard deviation as the criterion for best path selection less than 20% of the paths over both protocols have 20 *ms* or larger increases in standard deviation compared to that of the best path (with the lowest standard deviation of RTTs) of corresponding trace timelines.

Figure 6 offers a different perspective to understanding the impact of routing changes on the end-to-end RTTs of server-to-server paths. We pick three different thresholds–$20\,\text{ms}$, $50\,\text{ms}$, and $100\,\text{ms}$–each denoting the least value by which the end-to-end RTT between a server pair was increased due to routing changes (resulting in sub-optimal AS paths). We identify, for each trace timeline, all the sub-optimal AS paths that increase the end-to-end RTT by at least a chosen threshold value, and compute, separately for each trace timeline, the sum of prevalence of these paths. Figure 6 shows the ECDFs of the prevalence of sub-optimal AS paths, one for each threshold, over both IPv4 and IPv6.

Figure 6 offers further evidence that typically a routing change causes only a small change in RTT; the duration (or prevalence) of the sub-optimal path is often short. Figure 6 also shows that for a minority of cases the change can be significant. Looking at the tail of the distributions, for 10% of trace timelines over IPv4 the (sub-optimal) AS paths that led to at least a $20\,\text{ms}$ increase in RTTs had a prevalence of at least 30%, i. e., pertained for at least 30% of the time. In case of IPv6, the prevalence was at least 50%. For 1.1% of trace timelines over IPv4 and 1.3% over IPv6, sub-optimal AS paths resulting in at least $100\,\text{ms}$ increase in RTT had a prevalence of at least 20% over IPv4 and 40% over IPv6.

4.3 Data Trends over Short-term

In the previous section, we showed that most server pairs encountered relatively few AS-path changes, and that most of these changes did not significantly affect the end-to-end RTT of server-to-server paths. Although a small fraction of AS paths contributed to an increase of more than $20\,\text{ms}$ in the end-to-end RTTs, these paths were still relatively short-lived. We acknowledge, nevertheless, that our analyses can



(a) 10^{th} *percentile.* (b) 90^{th} *percentile.*

Figure 7: *Comparing magnitudes of increase in the (a)* 10^{th}*, and (b)* 90^{th} *percentiles of RTTs of AS paths (each relative to the "best" AS path of the corresponding trace timeline) in the short-term data set. Results using traceroutes conducted* 3 *hours apart (lines with suffix '3hr') are not different from that using traceroutes done* 30 *minutes apart (lines with suffix 'All').*

underestimate the number of AS path changes since the measurements are only available at 3-hour intervals. Routing changes can happen more frequently than every 3-hours, and such *missed* events can affect the inferences made. To assess the possible impact of the coarse granularity of measurements on the inferences drawn from the long-term data set, we turn to a subset of our short-term data set where measurements are made at a comparatively finer granularity. We used nearly $60M$ traceroutes gathered every 30 minutes, over a period of 22 days, from March 10, 2015 to March 31, 2015, between approximately $20K$ servers.

For each trace timeline, we paired the end-to-end RTTs of each traceroute with the AS path revealed by the traceroute. As explained in Section 4.2, we compute the deviations in the 10^{th} and 90^{th} percentiles of RTTs of each AS path of all trace timelines from that of the best AS path of corresponding timelines. To assess the possible impact of the coarse granularity of measurements, we repeat the computation, but consider only a subset of traceroutes separated by at least 3 hours in time. Figure 7 shows two ECDFs of the deviations in the 10^{th} and 90^{th} percentiles of RTTs associated with different AS paths: (a) the dotted lines, labeled 'IPv4 All' and 'IPv6 All', refer to the deviations computed by considering all traceroutes, and (b) the solid lines, labeled 'IPv4 3hr' and 'IPv6 3hr', refer to the deviations computed by only considering the subset of traceroutes separated by at least 3 hours. The difference between the 'All' and '3hr' ECDFs of each protocol is very small, indicating that the events that happen at shorter time scales likely did not affect the overall analysis in Section 4.2.

5. IMPACT OF CONGESTION

Is congestion the norm in the Internet core? Can we quantify the impact of congestion on the end-to-end RTTs of paths between servers? — these are the questions we examine in this section.

5.1 Is Congestion the Norm in the Core?

We seek to identify *consistent congestion*, defined as a type of congestion that has a *diurnal* cycle with each instance of congestion lasting for a few hours, and quantify its impact on server-to-server communication. We utilize the *Time Sequence Latency Probes (TSLP)* method and the *automated trace processing* technique using Fast Fourier Transform (FFT) described in [27] as follows. We apply the FFT (with frequency $f = 1/day$) on the time series of end-to-end RTTs between server pairs (as opposed to the difference in RTTs on successive hops as done in [27]). We consider only the pairs where the *power spectral density* (i. e., the power signal distribution around the frequency f) is significant. We divide the power at frequency f by the total power, and consider server pairs where this ratio is at least 0.3[2] (indication of strong diurnal pattern as a significant fraction of the energy is concentrated around the 24-hour period). The above is a simplification of the code from [27] which also detected short-lived daily variations and marginal cases. In the rest of the section, the word congestion refers only to consistent congestion, unless otherwise mentioned.

We analyzed server-to-server *ping* measurements collected using the CDN for a period of one week[3], from February 22 through 28, 2015. Servers located in one of approximately $10K$ clusters around the world pinged a pre-selected set of servers in other clusters every 15 minutes. The data set contains more than $2.9M$ IPv4 server pairs and more than $1M$ IPv6 server pairs with each pair having at least 600 (of the 672 possible) ping measurements. We calculated the difference between the 95^{th} and 5^{th} percentiles of RTTs per server pair to estimate the fraction of server pairs that experienced variations in RTTs exceeding $10\,\text{ms}$. Less than 9.5% of the server pairs over IPv4 and less than 4% over IPv6 observed more than $10\,\text{ms}$ of variation in RTTs in the one-week pe-

[2]The choice of the threshold was based on empirical evidence. We manually inspected some of the diurnal patterns in RTTs between server pairs captured at different thresholds, and settled on 0.3 as it seemed to capture the type of consistent congestion we wanted to investigate.

[3]We acknowledge that this is a brief duration, but we manually examined a sampling of the plots of RTTs to confirm the existence of diurnal patterns in each day of the week.

riod. When we consider server pairs with a strong diurnal pattern that experience such large variations in RTTs, the percentages of server pairs over IPv4 and IPv6 drops to 2% and 0.6% respectively. This indicates that consistent congestion is not the *norm* in the core of the Internet; one explanation, perhaps, is that links in the core are very often well provisioned, since congestion in the core may affect thousands, or even millions, of end-users. Note, however, that peering disputes have been reported to introduce long-term congestion in the Internet core [27] and a large population of end-users.

5.2 Locating Congestion

Unfortunately, with the ping data it is not possible to infer the congested router-level links. To infer the congested links we perform traceroute campaigns utilizing, both as a vantage point as well as a target, the subset of servers for which we have evidence that the path experienced congestion, as described in the previous section. The traceroute campaigns took place immediately after the ping campaign and lasted for three weeks. To ensure that a traceroute campaign completes before the next one is scheduled, we set the frequency of campaigns to 30 minutes.

We define the path from the vantage point of a traceroute to a given hop as a *segment*. Moving from the first hop towards the last hop (destination) of a traceroute, each segment of the traceroute contains the previous segment in its entirety and adds one more hop to it. To identify the segment of a traceroute where congestion occurs, we find the first segment that contributed to the overall increase in RTT between the endpoints of the traceroute. To reduce the chance that we did not isolate the contribution of congestion from other dynamics such as routing asymmetry and routing changes, we considered only the server pairs where the AS-level paths between them is symmetric and the IP-level path is static in each direction.

The algorithm works as follows. First, we create a time series of RTTs for each segment of the traceroute, and for each direction. We then re-calculate the FFT for each time series, with the frequency and threshold as described in the previous section. For more than 30% of the IPv4 and IPv6 server pairs with consistent congestion, we observe that a strong congestion signal was present even weeks after the initial observation. We then infer, for these server pairs, which time series of which segment best matches the pattern of the end-to-end server pair delay variation in each direction; a match implies that the corresponding segment contributes most towards the increase in RTT between the associated server pairs.

To compare the time series of RTTs of each segment, on a path between a server pair, with the time series of RTTs between the server pair, we use the *Pearson correlation coefficient (ρ)*. ρ takes values between -1 and 1, with the higher values denoting a higher similarity between the two compared time series. We set the threshold to $\rho = 0.5$ to select the segment of a traceroute responsible for congestion between the endpoints. An important insight is that once we infer the first segment of the traceroute that exceeds the threshold the following segments have similar or higher values; this is expected as the RTT level of the links in these segments has been increased. For our results, we mark the first segment in each direction as the congested link. Thus, it is possible to locate congestion with high confidence when it is found at the same link (or AS) on both directions of the path. We are, however, aware of the complications of traceroutes [39], and, thus, can only indicate if the congestion occurs inside a network or at an interconnection of the two networks. In the next section we describe how we distinguish between internal and interconnection links.

5.3 Identifying Router Ownership

Inferring whether a given link is an internal AS link or an interconnection is not trivial. Our method utilizes an IP-to-AS mapping derived from BGP data as well as CAIDA's AS relationship inferences from the same BGP data [29]. Because we may observe an IP address where the origin AS of the longest matching prefix in BGP maps to AS_x on a router operated by AS_y when crossing an interdomain link, we devise a set of heuristics to annotate AS owners of routers building on the initial AS mapping provided by BGP. Because inference of a router's ownership is impacted by ambiguities in traceroute (such as third-party IP addresses [45]) and flaws in IP-to-AS mappings [32, 45, 22], our method annotates the *likely* owner of most, but not all interfaces. In addition, the traceroutes we use only cover paths between servers, and thus, we lack visibility of other paths that may have provided constraints that enable us to infer ownership of more routers, and stress the need for an approach that has been thoroughly validated.

We processed all traceroute paths as a set focusing on sequences of IP address hops $IP_x, IP_y, \ldots, IP_z$ revealed in the traceroute paths, where each hop is a different IP address and not in the IANA reserved range. Our process begins by labeling each IP address with the *possible* inferences of ownership based on other IP addresses surrounding it in traceroute paths. Figure 8 illustrates five of the six heuristics we used and lists the names of heuristics. For instance, if both IP_x and IP_y were announced by AS_i, we labeled IP_x as being possibly owned by AS_i as in Figure 8a. We refer to this heuristic as the *first* heuristic since IP_x appeared before IP_y in the traceroute, and both addresses were announced by AS_i. Similarly, suppose IP_y does not have an IP-to-AS mapping because the address was not announced in BGP, but the IP addresses IP_x and IP_z at the surrounding hops were announced by AS_i as in Figure 8b; in this scenario, we labeled IP_y as being possibly owned by AS_i.

We also used the AS relationship inferences to inform our ownership inferences as follows. If IP_x and IP_y are announced by AS_i and IP_z by AS_j, and AS_j is a customer of AS_i, then we labeled IP_y as on a router possibly owned by AS_j based on the heuristic that, to interconnect with a provider, a customer typically uses the address assigned by that provider, as in Figure 8c. Similarly, not shown in Figure 8, if IP_x is announced by AS_i and IP_y by AS_j, and AS_j is a provider of AS_i, then we labeled IP_y as on a router possibly owned by AS_j using the heuristic that we observed the address on an interface of a provider's router that is facing

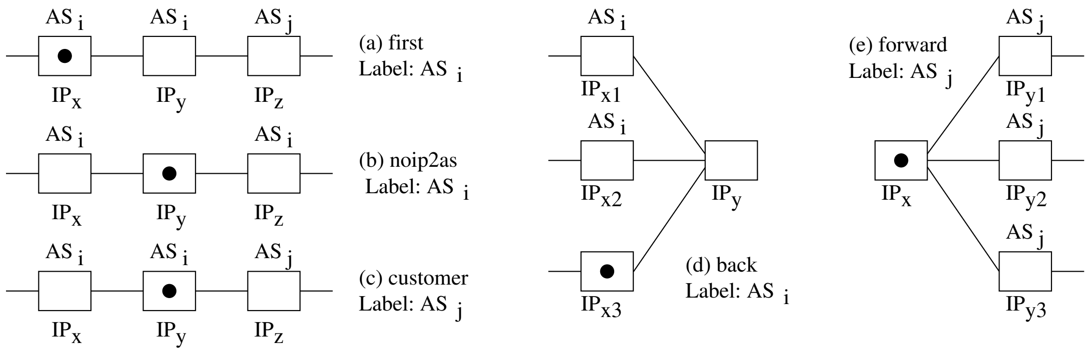


Figure 8: *Router ownership inference heuristics used on traceroutes to label interfaces observed with possible owners. The interfaces being labeled are noted with a dot.*

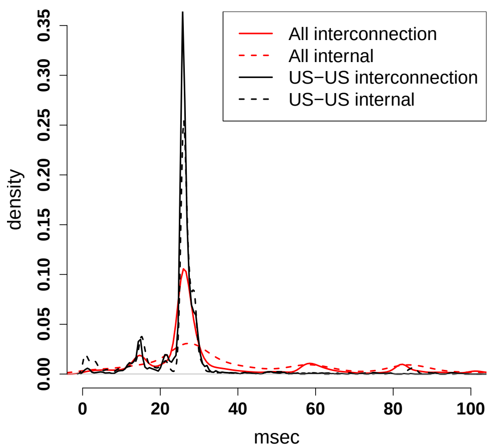


Figure 9: *Density of the congestion overhead in our dataset.*

its customer.

Because of the sparsity of our traceroute data, some IP addresses are observed in paths with insufficient constraints to make an inference. For example, there is no convention in a peering relationship that suggests which AS provides the address space used to establish a point-to-point link. If we had traceroute paths that observed the same addresses towards other networks, these paths could provide additional labels on who the likely owner of the router is. However, when we examined these addresses, they were often observed adjacent to other addresses for which we had possible labels. We exploit this insight as follows. Suppose we observed links $IP_{x1} - IP_y$, $IP_{x2} - IP_y$, and $IP_{x3} - IP_y$, and we had labeled IP_{x1} and IP_{x2} with the same possible AS owner AS_i as in Figure 8d; we then also label IP_{x3} as also possibly owned by AS_i provided that AS_i also announces the router's interface in BGP, with the assumption that traceroute paths to other destinations might have fulfilled the *first* constraint as with IP_{x1} and IP_{x2}. Similarly, suppose all links observed from unlabeled IP_x were $IP_x - IP_{y1}$, $IP_x - IP_{y2}$, and $IP_x - IP_{y3}$, and IP_{y1}, IP_{y2} and IP_{y3} were all mapped to AS_j and all had owner labels as in Figure 8e; we label IP_x as being possibly owned by the AS announcing IP_{y1}, IP_{y2} and IP_{y3} in BGP.

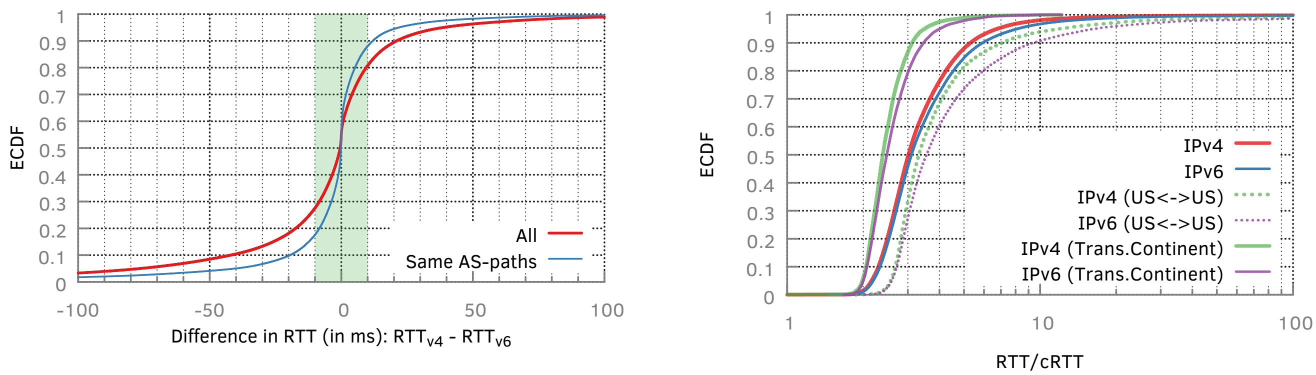
Once we have labeled IP addresses with their possible AS owners, we infer, for each IP address, one AS (as the owner) from the candidate owners available for that address. For addresses with only one possible candidate, the algorithm trivially assigns that candidate as the sole owner. When an address has multiple candidate owners, if the most frequent label applied was the *first* heuristic we use that corresponding AS owner. Note that our approach is solving a different problem than the heuristics proposed by Chen *et. al* [15]. Their work was focused on accurately inferring AS links from traceroute paths, rather than inferring which AS operates a given router.

With the inferences of AS owners of router IP addresses, we can then infer whether a congested link is an *internal link* or *interconnection* and infer the link type viz., *provider-to-provider (p2p)* and *customer-to-provider (c2p)*. Because traceroutes from more than one server pair can traverse the same link, it is possible for a link to be marked as congested by more than one server-to-server pair - in some cases hundreds of server-to-server pairs.

In our study of IPv4 traceroutes we investigated around 310K IP-IP links in total. We identified 3155 IP-IP links to be responsible for consistent congestion (congestion that has a diurnal cycle with each instance of congestion lasting for a few hours). Of these around 1768 were internal links and 1121 were interconnection links. Note, however, that it is possible to misidentify a number of interconnection links as internal. Of the 1121 interconnection links, we identified 658 to be p2p and 463 to be c2p. For the remaining 266 links it was not possible to infer if the link was an internal or interconnection link. Thus a higher number of internal links are detected to be congested as compared with interconnection links. However, when we weight the links by the number of server-to-servers paths that cross them, we find that the interconnection links are more popular. The large majority of the interconnection links with congestion were private interconnects. In our datasets around 60 links that were established over the public switching fabric of IXPs experienced congestion. This is not surprising since most IXPs have strict service level agreements (SLAs) concerning the peak utilization level and duration of the switch port used by the members; for instance, refer to the SLA of one of the largest IXPs AMS-IX in Amsterdam [1]. Thus, we expect that significant congestion occurs in cross-connects.

5.4 Estimating Congestion Overhead

In Figure 9 we present the overhead due to the congestion



(a) *End-to-end RTTs over IPv4 and IPv6.*

(b) *Inflation in RTT.*

Figure 10: *IPv4 vs IPv6: (a) Comparison of RTT between endpoint pairs; shaded area implies where the RTT difference is insignificant, (b) Comparison of inflation in RTT; inflation over transcontinental links is lower compared to the rest.*

in our data set. For both internal and interconnection links the typical overhead is between $20\,\mathrm{ms}$ and $30\,\mathrm{ms}$. Indeed, values in this range contribute to more than 60% of the density for both types of links. A closer investigation shows that for server pairs within the US, values between $20\,\mathrm{ms}$ and $30\,\mathrm{ms}$ were responsible for close to 90% of the density. We attribute this to the uniform way that router buffers are configured with rule-of-thumb value of RTT to be $100\,\mathrm{ms}$. In Europe and Asia, the density around the range of $20\,\mathrm{ms}$ and $30\,\mathrm{ms}$ is less prominent – close to 30%. We attribute this to different buffer configurations used by operators, possibly due to the differences in RTTs that may be related to the size of the countries.

When focusing on trans-continental links the distribution shifts to higher values, typically around $60\,\mathrm{ms}$. We attribute this to higher buffer sizes due to high RTT due to trans-continental distances. In a number of interconnection links in Asia as well as in trans-continental links between Asia and Europe we noticed very high values – around $90\,\mathrm{ms}$. One explanation is that there may be multiple congested links in the path that cannot be inferred with traceroute, e. g., due to MPLS tunneling. However, from our data set, there is no clear correlation between the geographical distance or distance in IP hops between two regions and the overhead due to congestion. In fact, the magnitude of congestion is greater at some closely located server pairs as compared with distantly located server pairs. Inferences from IPv6 data set are very similar.

6. IPv4 vs IPv6

In the long-term data set, whenever we observe end-to-end RTTs in traceroutes between any pair of source and destination IP addresses, conducted at the same time, over both IPv4 and IPv6, we calculate the difference in measured RTTs over both protocols ($RTT_{v4} - RTT_{v6}$); the ECDFs of the RTT differences is shown in Figure 10a. The differences in RTTs observed from $826M$ traceroutes (corresponding to approx. $196K$ server pairs), show by the line labeled 'All' in Figure 10a, highlight that for nearly 50% of the traceroutes (shaded region) the end-to-end RTTs are similar (differences in end-to-end RTTs are less than $10\,\mathrm{ms}$) over both the protocols. For the remaining traceroutes, simply switching from one protocol to the other offers a reduction of more than $10\,\mathrm{ms}$ in latency. Looking at the tail above $50\,\mathrm{ms}$ (below $-50\,\mathrm{ms}$) shows that for 3.7% (8.5%) of the endpoint pairs, the RTT can be reduced by at least 50 ms by using IPv6 instead of IPv4 (IPv4 instead of IPv6).

We also looked at a subset of traceroutes ($170M$ traceroutes corresponding to $161K$ server pairs) that had similar AS paths over IPv4 and IPv6, and the ECDF of the RTT differences for this subset is also included in Figure 10a (line labeled 'Same AS-paths'). Even in this restricted set, the choice of protocol is relevant for approximately 30% of the traceroutes (RTT differences are at least $10\,\mathrm{ms}$). Similar to the trend observed in the previous unrestricted case, RTTs over IPv4 are slightly lower compared to that over IPv6; while IPv6 offers $10\,\mathrm{ms}$ or lesser RTT compared to IPv4 for 10% of the traceroutes, RTTs over IPv6 is higher compared to that over IPv4 by at least $10\,\mathrm{ms}$ for more than 18% of the traceroutes. Overall performance between dual-stacked servers with the same AS path in both protocols result in much more similar delays than those that use different AS paths, in line with similar prior studies [33].

We have ground truth on the locations of the servers in the data set, and we analyzed traces from over $750K$ endpoint pairs and computed for each pair the median RTT. For each pair, we also calculated *cRTT*, defined as the time it takes for a packet traveling at the speed of light in free space to traverse the round-trip distance between the endpoint pair. We define *inflation* between an endpoint pair as the ratio of the median observed RTT to the cRTT of that endpoint pair. Figure 10b shows that in the median the observed inflation over IPv4 (3.01) and IPv6 (3.1) do not differ much, and even at the 90^{th} percentiles the difference (5.3 for IPv4 and 5.9 for IPv6) is minimal. These values are comparable to a similar study of inflation [42] but on the end-user to server paths. Figure 10b also shows inflation between endpoint pairs both of which are in US, and inflation where the path between endpoint pairs (connecting US and Germany, or US and Australia, or US and India, or US and Japan) involves transcontinental links. Not surprisingly, in our data, inflation involving transcontinental links is significantly lower compared to inflation between endpoints pairs in the US.

While Singla et al. [42] geolocated the end-users for computing the RTT inflation, which can affect the results (depending on the accuracy of the geolocation), we have ground truth on locations of servers, and we also compare the inflation in RTT between the endpoints over both IPv4 and IPv6. The inflation shown in Figure 10b is based only on end-to-end RTTs between servers and hence, inflation in the core most likely stem from infrastructure inefficiencies; for instance, there are no DNS lookups or TCP handshakes involved in these calculations.

7. DISCUSSION

Using server-to-server paths as a proxy to the Internet's core, we show that congestion is not the norm in the Internet's core and also that routing changes, for the most part, do not significantly affect the end-to-end RTTs. Studies [46] have shown that when the RTTs of end-user to server paths increase significantly, a significant fraction of such instances can be attributed to routing changes. In the context of server-to-server paths, our results confirm, however, that the reverse is not true: routing changes only rarely have a significant impact on the end-to-end RTTs of paths.

A natural question to ask is whether the non-typical cases are more prevalent for routing or congestion. Although we do not have an ideal side-by-side comparison, our results suggest that routing changes have the greater impact. Figure 6 on routing shows that for 10% of trace timelines the (sub-optimal) AS paths that led to at least $20\,\mathrm{ms}$ increase in RTTs pertained for at least 30% of the study period for IPv4 and 50% for IPv6. In contrast, in Section 5 on congestion, just 2% (much less than 10%) of the server pairs over IPv4, and just 0.6% over IPv6, experience a strong diurnal pattern with an increase in RTT of least $10\,\mathrm{ms}$. It is, nevertheless, possible that in case of a particular server pair congestion might be the greater issue.

The observed performance characteristics may be due to the structural changes in Internet's core, also referred to as "flattening of the Internet" [39, 18, 5], or the increase of peering locations [12, 13], or the pressure on transit providers to offer cheaper and competitive service [26], or investments in alternative or parallel high-speed networks [42], or the deployment of the massively distributed server infrastructures [10, 23, 34, 2, 6]; it is hard to pin-point the exact root cause without further studies.

Studies of known peering disputes [27] show that congestion of particular links that run "hot" experience unusually high delays, but the congestion disappears in a few hours after agreements to settle the disputes. These emphasize the highly variable nature of the Internet's core necessitating continuous monitoring over longer time periods for accurate evaluations. Our work is an example of such a continuous monitoring over a period of 16-months.

An understandable criticism is that the view of the Internet's core from the perspective of a service provider may be different than that of other networks, e. g., eyeball, transit. Indeed, it is possible that measurements from other platforms, viz., Dasu [40], iPlane [31], may offer a different view of the Internet's core. Such studies can complement our work and aid in finding the exact root causes behind the observations.

8. CONCLUSION

In this paper, we exploit a large corpus of measurements in both forward and reverse directions between thousands of CDN servers to offer insights into the performance characteristics of the Internet's core. Our data suggests that significant daily oscillations in latency, herein called congestion, is not the norm in the Internet's core. However, at times, some links do experience congestion, and we detected such incidents on a greater number of internal links than interconnection links. When we weight the links by the number of server-to-servers paths that cross them, we find that the interconnection links are more popular. The large majority of the interconnection links with congestion were private interconnects. We also show that while routing changes typically have marginal or no impact on the end-to-end RTTs, 4% (7%) of routing changes on IPv4 (IPv6) increase RTTs by at least $50\,\mathrm{ms}$ for at least 20% of the study period. The non-typical cases affecting RTTs are more prevalent for routing than congestion.

This paper presents only a study of latencies in the Internet's core. We encourage follow-up work focusing on other characteristics, viz., available bandwidth, packet loss, and utilizing other measurement platforms to further enrich our understanding of the Internet's core. The similarity in performance characteristics over IPv4 and IPv6 also naturally calls for a study to understand to what extent infrastructure is shared between IPv4 and IPv6, and we plan on addressing this question in future work.

9. ACKNOWLEDGMENTS

This work was supported in part by NSF CNS-1414177. Georgios Smaragdakis was supported by the EU Marie Curie IOF "CDN-H" (PEOPLE-628441). Matthew Luckie conducted this work while at CAIDA, UC San Diego. We would also like to thank our shepherd Harsha Madhyastha, David Clark and Steve Bauer for their constructive feedback.

10. REFERENCES

[1] AMS-IX Service Level Agreement. https://ams-ix.net/services-pricing/service-level-agreement.

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

[3] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. L. Zhang. Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery. In *IEEE INFOCOM*, 2012.

[4] V. K. Adhikari, S. Jain, and Z. L. Zhang. YouTube traffic dynamics and its interplay with a tier-1 ISP: an ISP perspective. In *ACM IMC*, 2010.

[5] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *ACM SIGCOMM*, 2012.

[6] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Web Content Cartography. In *ACM IMC*, 2011.

[7] M. Allman. On Changing the Culture of Empirical Internet Assessment. *ACM CCR*, 43(3), 2013.

[8] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding Traceroute Anomalies with Paris Traceroute. In *ACM IMC*, 2006.

[9] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *ACM IMC*, 2009.

[10] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *ACM IMC*, 2013.

[11] H. Chang, S. Jamin, and W. Willinger. Inferring as-level internet topology from router-level path traces. In *Proceedings of SPIE ITCom*, 2001.

[12] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. There is More to IXPs than Meets the Eye. *ACM CCR*, 43(5), 2013.

[13] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. Quo vadis Open-IX? Trying to boost public peering in the US. *ACM CCR*, 45(1), 2015.

[14] F. Chen, R. K. Sitaraman, and M. Torres. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM*, 2015.

[15] K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao. Where the Sidewalk Ends: Extending the Internet AS Graph Using Traceroutes From P2P Users. In *CoNEXT*, 2009.

[16] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey. Measuring IPv6 Adoption. In *ACM SIGCOMM*, 2014.

[17] L. Deng and A. Kuzmanovic. Monitoring Persistently Congested Internet Links. In *IEEE ICNP*, 2008.

[18] A. Dhamdhere and C. Dovrolis. Twelve Years in the Evolution of the Internet Ecosystem. *IEEE/ACM Trans. Networking*, 19(5), 2011.

[19] F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica, and H. Zhang. Understanding the Impact of Video Quality on User Engagement. In *ACM SIGCOMM*, 2011.

[20] I. Drago, M. Mellia, M. Munafo, A. Sperotto, R. Sadre, and A. Pras. Inside Dropbox: Understanding Personal Cloud Storage Services. In *ACM IMC*, 2012.

[21] T. Holderbach, C. Pelsser, R. Bush, and L. Vanbever. Quantifying interference between measurements on the RIPE Atlas platform. In *ACM IMC*, 2015.

[22] Y. Hyun, A. Broido, and k. claffy. Traceroute and BGP AS Path Incongruities. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), Mar 2003.

[23] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart, and A. Vahdat. B4: Experience with a Globally-Deployed Software Defined WAN. In *ACM SIGCOMM*, 2013.

[24] A. Khan, T. Kwon, H. C. Kim, and Y. Choi. AS-level Topology Collection through Looking Glass Servers. In *ACM IMC*, 2013.

[25] A. Li, X. Yang, S. Kandula, and M. Zhang. CloudCmp: Comparing Public Cloud Providers. In *ACM IMC*, 2010.

[26] A. Lodhi, A. Dhamdhere, and C. Dovrolis. Open Peering by Internet Transit Providers: Peer Preference or Peer Pressure? In *IEEE INFOCOM*, 2014.

[27] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, and kc claffy. Challenges in Inferring Internet Interdomain Congestion. In *ACM IMC*, 2014.

[28] M. Luckie, A. Dhamdhere, k. claffy, and D. Murrell. Measured Impact of Crooked Traceroute. *ACM CCR*, 44(1), Jan 2011.

[29] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and kc claffy. AS Relationships, Customers Cones, and Validations. In *ACM IMC*, 2013.

[30] M. Dischinger and M. Marcon and S. Guha and K. P. Gummadi and R. Mahajan and S. Saroiu. Glasnost: Enabling End Users to Detect Traffic Differentiation. In *NSDI*, 2010.

[31] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. 2006.

[32] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an Accurate AS-level Traceroute Tool. In *ACM SIGCOMM*, 2003.

[33] M. Nikkhah, R. Guérin, Y. Lee, and R. Woundy. Assessing IPv6 through web access a measurement study and its findings. In *CoNEXT*, 2011.

[34] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS Oper. Syst. Rev.*, 2010.

[35] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Trans. Networking*, 5(5), 1997.

[36] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush. From Paris to Tokyo: On the Suitability of ping to Measure Latency. In *ACM IMC*, 2013.

[37] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery using Provider-aided Distance Information. In *ACM IMC*, 2010.

[38] E. Pujol, P. Richter, B. Chandrasekaran, G. Smaragdakis, A. Feldmann, B. Maggs, and K. C. Ng. Back-Office Web Traffic on The Internet. In *ACM IMC*, 2014.

[39] M. Roughan, W. Willinger, O. Maennel, D. Pertouli, and R. Bush. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *IEEE J. on Sel. Areas in Comm.*, 29(9), 2011.

[40] M. A. Sanchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger. Dasu: Pushing Experiments to the Internet's Edge. In *NSDI*, 2013.

[41] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratsanamy, and V. Sekar. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service. In *SIGCOMM*, 2012.

[42] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. M. Maggs. The Internet at the Speed of Light. In *SIGCOMM HotNets*, New York, NY, USA, 2014.

[43] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescape. Broadband Internet Performance: A View From the Gateway. In *ACM SIGCOMM*, 2011.

[44] S. Triukose, Z. Wen, and M. Rabinovich. Measuring a Commercial Content Delivery Network. In *WWW*, 2011.

[45] Y. Zhang, R. Oliveira, H. Zhang, and L. Zhang. Quantifying the pitfalls of traceroute in AS connectivity inference. In *PAM*, 2010.

[46] Y. Zhu, B. Helsley, J. Rexford, A. Siganporia, and S. Srinivasan. LatLong: Diagnosing Wide-Area Latency Changes for CDNs. *IEEE Tran. on Netw. and Serv. Manag.*, 43(9), 2012.