Inferring Persistent Interdomain Congestion

Amogh Dhamdhere, David D. Clark[†], Alexander Gamero-Garrido
Matthew Luckie[*], Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai[•]
Alex C. Snoeren and kc claffy

CAIDA/UC San Diego [†]MIT [*]University of Waikato [•]TU Munich

ABSTRACT

There is significant interest in the technical and policy communities regarding the extent, scope, and consumer harm of persistent interdomain congestion. We provide empirical grounding for discussions of interdomain congestion by developing a system and method to measure congestion on thousands of interdomain links without direct access to them. We implement a system based on the Time Series Latency Probes (TSLP) technique that identifies links with evidence of recurring congestion suggestive of an under-provisioned link. We deploy our system at 86 vantage points worldwide and show that congestion inferred using our lightweight TSLP method correlates with other metrics of interconnection performance impairment. We use our method to study interdomain links of eight large U.S. broadband access providers from March 2016 to December 2017, and validate our inferences against ground-truth traffic statistics from two of the providers. For the period of time over which we gathered measurements, we did not find evidence of widespread endemic congestion on interdomain links between access ISPs and directly connected transit and content providers, although some such links exhibited recurring congestion patterns. We describe limitations, open challenges, and a path toward the use of this method for large-scale third-party monitoring of the Internet interconnection ecosystem.

CCS CONCEPTS

• **Networks** → *Network measurement*;

KEYWORDS

Internet congestion, Internet topology, performance

ACM Reference Format:
Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and kc claffy. 2018. Inferring Persistent Interdomain Congestion. In *SIGCOMM '18: ACM SIGCOMM 2018 Conference, August 20–25, 2018, Budapest, Hungary*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3230543.3230549

SIGCOMM '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5567-4/18/08…$15.00
https://doi.org/10.1145/3230543.3230549

1 INTRODUCTION

In its strictest definition—demand exceeds capacity of a resource—congestion is a widespread phenomenon on the Internet, and central to the proper functioning of TCP. An endpoint of a TCP connection induces congestion as a means to ascertain its most appropriate sending rate, increasing its packet sending rate until it detects a failure of the other end to acknowledge receipt of a packet. Other common occurrences, including traffic management transitions, router operating system overheads, network configuration errors, flash crowds (e.g., software releases), and malicious attacks can induce isolated episodes of network congestion. These are inevitable and inherent aspects of packet switched public IP-based networks, and are not the focus of this work.

Our interest is in persistent congestion due to a long-term mismatch between installed capacity and actual traffic, particularly at points of interconnection between networks. In this context, congestion manifests as an increase in the latency of packet delivery due to the time that the packet waits in a router buffer, and potentially dropped packets, which may be an impairment itself but also causes a sender to slow its sending rate, and poses a risk to the user quality of experience (QoE). Although of regulatory interest in the U.S. for years [37], there have thus far been no methods or tools for execution of lightweight measurement from the edge that is capable of detecting this type of congestion.

Several recent peering disputes covered in the press [10, 17–19, 31, 32, 69] elicited various claims by involved parties regarding the causes of congestion and poor performance. However, there is a dearth of publicly available data that can shed light on interconnection issues. In isolated cases,

providers have volunteered limited, anonymized information that reveals congestion at interconnection points [66]. In 2014 and 2015, the Measurement Lab (M-Lab) consortium released reports that inferred congestion at interconnection points between large content and access networks in the U.S. using crowd-sourced Network Diagnostic Tests (NDT) by end users [9, 51]. Some content providers release end-to-end performance data (e.g., video quality reports) that provide aggregate information about performance between themselves and access providers [42, 54]. However, end-to-end statistics cannot accurately map congestion to specific interconnection points, and the coverage of crowd-sourced measurements is low [65]. In 2016, seven access providers volunteered anonymized and aggregated statistics of interdomain link utilization, but the level of aggregation prevented inference of congestion on individual links [35]. The U.S. broadband ecosystem (access and transit ISPs as well as edge providers) is at a daunting crossroads—without clear regulatory oversight, and without transparency into performance of critical components of the infrastructure.

We report the results of an effort in which we developed, validated, and operationalized a method to detect congestion at interconnection points between networks *at a link-level granularity*. Our system identifies all the interdomain links visible from a vantage point (VP) in a given access ISP [49], and uses latency measurements to infer congestion on those links [48]. This paper provides the following contributions:

(1) We describe the design and implementation of a system that conducts ongoing congestion measurements of thousands of interdomain links from a set of topologically and geographically diverse vantage points. Since March 2016, we have collected measurements from 86 vantage points in 47 ISPs. For every link we measure, we infer whether that link shows evidence of congestion, and report the duration that congestion was present.

(2) We validate our inferences of congestion using other approaches and with direct operator feedback. We show that inferences of congestion using the TSLP technique are consistent with those obtained from more invasive active measurements (loss rate, throughput, and video streaming performance). We also use direct operator validation to confirm that our specific inferences about congestion match their proprietary data on link utilization.

(3) We report results of a longitudinal measurement study of interdomain congestion in eight broadband access networks in the U.S. Our focus on the U.S. broadband ecosystem is motivated by a history of public peering disputes between large U.S. broadband access providers and content providers [10, 17–19, 31, 32, 69], and the concentration of our measurement VPs in the U.S. For the period of time over which we gathered measurements, we did not find evidence of widespread endemic congestion of interdomain links between U.S. access ISPs and directly connected transit and content providers. We did find evidence of congestion between specific parties, in some cases quite severe, e.g., exceeding half the day for many days.

(4) We are publicly releasing our analysis scripts, and the underlying datasets via an interactive visualization interface and query API to encourage reproducibility of our results. Our data management system, based on the InfluxDB [2] time-series database and Grafana [1] visualization front-end, allows interactive data exploration, near real-time views of interdomain links, and longitudinal views. While this paper focuses on data from U.S. broadband access providers, we are publicly releasing measurements from VPs outside the U.S. as well.

2 RELATED WORK

Interdomain interconnection is one of the classic tussles in the Internet [27], as networked organizations must interconnect to provide connectivity, but those networks may have misaligned interests. In 2009, Bauer et al. [16] presented an overview of the evolution of Internet congestion, arguing that recent increases in edge traffic demands and changes in user expectations have forced network operators to use non-TCP congestion control mechanisms, such as volume-based limits and active traffic management. At the same time, the Internet's interconnection structure has evolved [29, 41], most notably the proliferation of Internet Exchanges (IXes) as anchor points in the mesh of interconnection [7]. In 2010, Labovitz et al. [46] reported that the majority of interdomain traffic was exchanged directly between content and consumer networks, and by 2013 Netflix accounted for a third of peak downstream traffic [63]. These changes have resulted in heated disputes (e.g. [10, 17–19, 31, 32, 69]) over the provision and management of interdomain links, as no single operator has complete control of the link's operation. Characterizing these disputes is challenging due to scant publicly available data on interconnection.

In 2008, Deng et al. [28] introduced a method for detecting interdomain congestion using latency measurements. Because their *pong* tool requires cooperative endpoints, i.e., a system on the other side of an interdomain link, the approach is challenging to deploy at scale. In 2014, Luckie et al. [48] introduced the Time Series Latency Probes (TSLP) technique, which sends probe packets toward an address across a link, with TTLs set to expire at the near and far routers attached to the link. The TSLP approach is appealing because of its lightweight nature and the feasibility of implementing measurements from the edge of the network, without cooperation from operators or direct access to border routers (unlike [30]). Luckie et al. [48] identified several

challenges in applying TSLP at scale, including the topological problem of accurately identifying interdomain links, and that replies might travel along an asymmetric reverse path. While they evaluated TSLP on a single operational link from a research network, that link did not have persistent congestion, and they did not conduct systematic validation.

In 2016, Luckie et al. [49] tackled the interdomain border-mapping challenge, developing and validating the *bdrmap* algorithm to infer interdomain links of an access provider *at the IP-link level granularity*. This work expanded on the then state-of-the-art in characterizing interdomain (AS-level) connectivity [24, 56–58, 62]. Recently, Fanou et al. [33] used *bdrmap* and TSLP to detect and analyze the causes of congestion at IXP infrastructure in Africa. Chandrasekaran et al. [23] and Fontugne et al. [39] used traceroute data — between pairs of CDN servers and from RIPE Atlas, respectively – to analyze routing changes, forwarding anomalies, and elevated latencies due to congestion. Both efforts used existing traceroute measurements to characterize observed paths. In contrast, our work uses focused measurements targeted at all visible interdomain links of the network hosting our VPs, and we conducted systematic validation.

Work on characterizing interdomain congestion in the U.S. is sparse, largely due to the proprietary nature of the data required to validate inferences. Different stakeholders are willing to publish reports, aggregated data, or end-to-end throughput measurements, but none of these approaches allow insight into individual interconnections. The Measurement Lab studies [9, 51] did not have path information, without which it is challenging to conclude that observed congestion is at the interconnection (it could be internal to ASes). Futhermore, the crowd-sourced nature of NDT tests used in these studies make it difficult to conclude that observed diurnal variations are due to congestion as opposed to variations due to a different testing sample [65]. In 2017, Sundaresan et al. [64] proposed a technique that uses TCP connection statistics to determine if a TCP flow experienced *self-induced* or *external* congestion. While the technique provides more information than a regular speed test about what limits a TCP flow, it still cannot localize *where* the bottleneck lies.

The methods we use in this paper enable fine-grained link-level congestion inferences, which can be aggregated to higher levels of granularity such as region and provider-wide. We hope that our systematic validation of the method yields an opportunity for the FCC and other regulators, as well as third-parties, to augment existing measurement fabrics with a lightweight method for capturing potential interconnection performance issues.

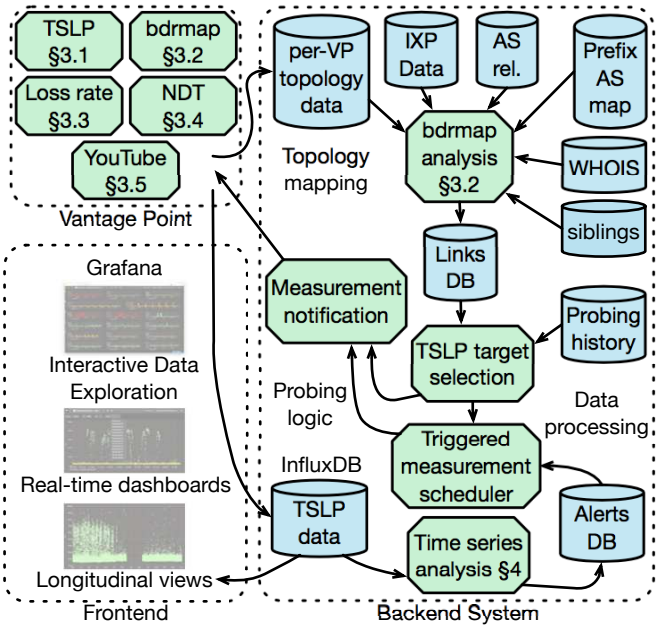


Figure 1: Our system for interdomain link discovery, active measurements, and congestion inference.

3 SYSTEM AND METHODS

Figure 1 provides an overview of the measurement system we built to operationalize the TSLP method and execute it at scale. The system utilizes a set of Vantage Points (VPs) to execute a diverse set of measurements: TSLP (§3.1) and *bdrmap* (§3.2) measurements which form the core of the method, and three separate measurement efforts to validate the TSLP method and to provide context on measurable impacts of congestion: high-frequency loss measurements (§3.3), throughput measurements (§3.4), and video streaming performance tests (§3.5). The backend system is responsible for maintaining an up-to-date TSLP probing state and managing time-series data, for which we use InfluxDB [2]. We have set up a Grafana [1] frontend to provide interactive visualization of our data via a web browser. Since March 2016, our system has used 86 VPs in 47 networks in 24 countries. Due to the volunteer-based nature of Ark VP hosting, there is churn in the set of usable VPs. As of December 2017, our measurements spanned 63 VPs in 39 networks in 22 countries. In §6 we report on results from 29 VPs in 8 broadband access networks in the United States. To encourage reproducibility of our results, we will provide public access to the Grafana interface and API access to the data in InfluxDB.

3.1 Time-Series Latency Probing method

The *Time-Series Latency Probing (TSLP)* method [48] builds on a basic intuition: if the offered load at a link approaches (or exceeds) capacity, then packets are buffered, leading to an increase in measured latency through that link. Given a

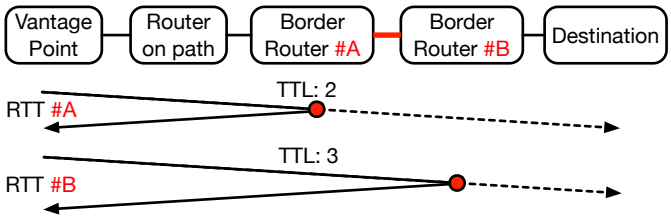


Figure 2: The Time-series latency probing (TSLP) method sends TTL-limited packets expiring at border routers #A and #B to measure link delay patterns.

vantage point (VP) inside a network, we use ICMP probes to measure latencies over time to the two ends of an identified interdomain link (*near* and *far* end, BR #*A* and BR #*B* in Figure 2). If the latency to the far end of the link is elevated but that to the near end is not, then a possible cause of the increased latency is congestion at the interdomain link. Because packets sent directly *to* routers are often processed differently (e.g., in a less-optimized code path) from those sent *through* them, TSLP sends ICMP probes to a destination *behind* the target link, with TTLs such that the packets expire at the near and far interfaces.

For each inferred interdomain link (§3.2), the TSLP target selection module identifies up to three destinations such that both the near and far ends of the target link are on the forward path toward those destinations. We prefer destinations in the address space of the neighbor network; however, this is not possible for all inferred interdomain links. For each measured link, TSLP uses all three destinations to probe both endpoints of the link every five minutes. While probing the near and far endpoints of a link using the same destination, we maintain the flow identifier of our probes constant by using the same ICMP checksum, so as to force the same forward path toward the link in the presence of per-flow load balancing (ECMP) [13]. Additionally, since June 2017 we have implemented a feature that uses the same flow identifiers across all *bdrmap* runs used to discover interdomain links (§3.2). Using three destinations provides some redundancy in case the route toward a destination changes to no longer traverse the target link. When periodically updating the probing set (§3.2), we do not change the destination used to probe a link unless that destination has lost visibility of the link. This ensures that over time the set of destinations (and thus the forward paths to the link through the VP's network) are constant to the extent possible. To minimize load on the VP's host, each VP limits its overall probing TSLP rate to 100 packets per second (pps).

3.2 *bdrmap*: Identifying links to probe

Each VP runs *bdrmap* [49] to infer the interdomain links between the host network and neighbor networks visible from the VP. The interdomain links that *bdrmap* infers represent IP-level links between the border routers of the host network and neighbor network. *bdrmap* uses an efficient variant of traceroute to trace the path to every routed prefix observed in BGP, and also performs alias resolution measurements on the set of discovered interfaces (using Ally and Mercator, see [49] for details). The border mapping data collection runs continuously in the background on the VP at a low probing rate (100 packets per second). The *bdrmap* analysis module processes this raw topology data, applying a set of heuristics [49] to infer interdomain links of the network hosting the VP. It uses as input a prefix-to-AS mapping constructed from public BGP data (RouteViews [55] and RIPE RIS [5]), a set of AS-relationships from CAIDA's AS-relationship algorithm [20, 50], a list of IXP prefixes curated from Packet Clearing House (PCH) [59] and peeringDB [4] data, WHOIS data from RIR delegation files [6, 11, 12, 47, 61], and a list of sibling ASes of the network hosting the VP.

We create sibling lists based on CAIDA's AS-to-organization dataset [21], which is generated quarterly by automatically parsing WHOIS data. However, automatic processing of WHOIS data is inherently error-prone, and we need an accurate list of siblings of the network hosting the VP, or we risk mis-identifying the boundaries of the organization corresponding to the hosting network. We undertook a manual effort, reviewing WHOIS information to add missing siblings of the network hosting the VP and remove spurious ones. We are releasing our list of AS siblings used for this study along with other datasets and code.

Over time, the interdomain links visible from a VP, as well as the interdomain link traversed from a VP toward a destination may change. To keep the probing set up-to-date, we use the *bdrmap* traceroutes to continuously update the mapping between destinations and visible interdomain links. Currently, we use a full cycle of *bdrmap* traceroutes (which takes approximately 1–3 days depending on the monitor) to update the probing set. We found that this process is effective at keeping the probing set up to date: the response rate to our TSLP probes was greater than 90% for many of our VPs. We may still incur delays up to three days in correcting the loss of visibility due to a routing change in the network. In future work, we plan to make the probing update process reactive, updating the probing set as soon as we observe that a destination has lost visibility of an interdomain link.

3.3 Packet loss measurement

As with TSLP latency measurements, the loss measurement module sends TTL-limited ICMP echo probes toward both the near and far ends of interdomain links, set to expire at the target interfaces. Using an overall probing budget of 150pps from each VP, this module probes each target

interface every second. This probing rate generates 300 data points per link in a 5-minute window, in contrast to the TSLP probing described above, which generates 1-3 data points per link in a 5-minute window. Due to the high probing rate required to sample all interfaces once per second, this module performs loss measurements on only a subset of links probed with TSLP, specifically those that satisfy two conditions: (1) they interconnect to networks that are either providers or peers (not customers) of the network hosting the VP, as inferred by CAIDA's AS-relationship algorithm [20, 50], or to networks on a static list of ASes representing large transit and content providers; and (2) in a previous week they experienced at least one episode of congestion as inferred by one of the two methods described in §4.

3.4 Throughput measurements

We use the Network Diagnostic Tool (NDT) to measure upload and download TCP throughput to M-Lab NDT servers [3]. Given the invasive nature of the NDT test, and to ensure that its results can be used for validation, we identify appropriate M-lab NDT servers using traceroutes from VPs to NDT servers so that the tested path crosses an interdomain link that shows evidence of congestion. The NDT measurement performs upload and download throughput tests from the selected NDT server, each for 10 seconds. After the throughput tests, this module performs a traceroute toward the NDT server to obtain the forward path from the VP to the NDT server, from which it identifies the interdomain link traversed in the forward path. We also use traceroutes from the NDT servers toward our VPs, which the M-lab platform performs toward every client that conducts an NDT throughput test, for visibility into the reverse path.

3.5 YouTube streaming measurements

We measure YouTube streaming performance by downloading YouTube videos using the `YouTube-test` tool [8]. The tool first downloads the webpage of a given video to extract the video's manifest, which contains metadata such as the video/audio bitrate, encoding scheme, and URL for downloading the video from the selected cache. It then streams the video with the highest supported bitrate. We selected popular videos, at least 1-minute long. To evaluate video streaming performance, the tool emulates the playback process by buffering and decoding the video data. After the test completes, we perform a traceroute toward the IP address of the video cache to obtain the forward path. We then correlate the YouTube data collected from the VPs to the interdomain links by using the traceroute collected during the tests to match the hops to the near and far IPs of the links as seen from the VPs using *bdrmap*. Although asymmetric paths are possible during these tests, most connections to Google are via a front end across a direct peering link [22, 25], so the video traffic likely crosses that peering link.

Both NDT throughput tests and YouTube streaming tests are invasive measurements, so we perform them at low frequency—every 15 minutes from 5pm to 11pm local time at the VP, and hourly at other times.

4 CONGESTION INFERENCE

We next present two time-series analysis methods we used to infer congestion. The first, *level-shift*, analyzes time series in weekly chunks to find episodes of elevated latency (§4.1). We used this method for nine months to trigger reactive loss rate probing (3.3) from March to December 2017. We developed a second method that uses *autocorrelation* (§4.2) on a longer time window and is more effective at identifying links that show evidence of consistently recurring congestion, which are the primary focus of this work. Our system uses these methods to identify congestion windows represented as start and end timestamps of each inferred congestion event.

4.1 Level-shift

The level-shift detection heuristic is based on CUSUM [67]. As a pre-processing step, we select the minimum latency in a time bin to filter outliers, e.g., due to slow ICMP responses from target routers or transient queuing. Given a parameter l (termed the *cut-off length*), the algorithm detects level-shifts of duration at least $l/2$. The algorithm first estimates the average variance σ_l^2 of the entire time series, calculated as the average variance in a moving window of length l. It then determines the minimum difference Δ between the means of two adjacent regimes of length l that is statistically significant according to the Student's t-test (at the 95% confidence level), to infer a level-shift. To handle outliers in the time series, the algorithm employs Huber's weight function [43] with an adjustable parameter P where higher values of P accommodate more deviation, e.g., P=5 tolerates outliers up to 5 standard deviations. We use the algorithm with l=12 and P=1 (each data point representing the minimum latency in a 5-minute bin), i.e., the algorithm detects level shifts lasting at least 30 minutes.

4.2 Autocorrelation method

Autocorrelation finds patterns of similarity between elements of a time series separated by a repeating interval, in this case 24 hours. We use autocorrelation to find multi-day repetition of elevated delays at the same times of day that imply congestion driven by diurnal demand. An autocorrelation scheme must look at multiple days; we use a 50-day window.

The autocorrelation algorithm aggregates the raw TSLP measurements into 15-minute intervals. Our TSLP measurements capture latency to the near and far side of each link

in 5-minute intervals, for up to three destinations, yielding between three and nine measurements over a 15-minute interval. As with level-shift, the autocorrelation function selects the minimum latency in a 15-minute interval to filter outliers. We test for elevated latencies to the near side (which indicates possible congestion within the access network) and exclude those times so that we can focus on the interconnection link itself.

For each 15 minute interval in the day, we look across all 50 days in the current sample and count the number of days for which the RTTs in that interval were above our threshold of elevation, which we currently set at (min RTT + 7 ms). The more days that contribute elevated latency in the same interval, the more likely it is that the event that triggered the elevated RTTs is a recurring one. Once we find the interval where the most days have evidence of congestion, we then look for adjacent 15 minute intervals that also have sufficient elevated days, and declare that recurring congestion has occurred in that part of the day, which we call the window of recurring congestion.

To filter out false positives, we reject the hypothesis that a link manifests recurring diurnal congestion during the 50-day window if we cannot disambiguate candidate recurring congestion windows distributed across the day (as opposed to a single peak that suggests recurring diurnal congestion), or if different days contribute to different peaks. The algorithm then looks separately at each day, and counts the number of 15-minute intervals within that recurring congestion window in which the RTT is elevated above our threshold. The algorithm uses the number of elevated 15-minute intervals as the estimate of congestion on that day. For example, if a link had one elevated 15-minute interval on a day, the algorithm infers a congestion level of 1.04% (1/96).

To avoid making false inferences of congestion, we then manually inspect the results of the algorithm in cases where it asserts evidence of congestion, to confirm that the assertion is appropriate. If interdomain congestion was rampant, this task would be overwhelming, but in our data it was manageable. We manually examined all the asserted congestion in this paper as looking reasonable, in addition to the measurement-based validation of our methods (§5.4). The current tuning parameters of the autocorrelation algorithm are based on this experience examining traces and assertions.

The final stage of the scheme merges estimates from all VPs that observe a given interdomain link to derive an overall inference. Congestion inferences for the same link based on data from different VPs are typically similar. Significant differences may reflect an asymmetric return path (§7).

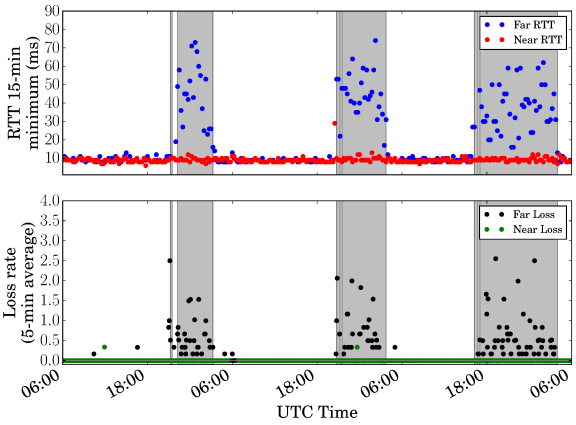


Figure 3: Time series of TSLP latency (top) and packet loss percentage (bottom) for an interdomain link between Verizon and Google on Dec. 7–9, 2017. Periods we infer as congested are shaded in gray.

5 VALIDATION

Given the politically loaded nature of interdomain congestion inferences, we undertook as many different approaches to validation as were feasible. We gathered three separate sources of data to validate our congestion inference methods: packet loss (§5.1), video streaming performance (§5.2), and throughput measurements (§5.3). We also obtained direct operator feedback from two large access ISPs (§5.4).

At a high level, our validation approach is to use the binary temporal classification of each 15-minute interval (either "congested" or "uncongested") produced by the autocorrelation algorithm to compare packet loss, YouTube streaming performance and NDT throughput statistics for the congested and uncongested periods.

5.1 Correlation with loss rate

We collected loss rate measurements using the reactive approach described in Section 3.3 from 15 Ark VPs from March–December 2017. In December 2017, after we had operationalized the autocorrelation technique (§4.2), we started a focused collection of loss data based on autocorrelation analysis of TSLP data from November 2017. We ran this data collection from 15 VPs from 12–31 December 2017.

To illustrate the correlation intuitively, Figure 3 shows TSLP and loss rate measurements for a link between Verizon and Google from Dec 7–9, 2017. The latency pattern to the far end was elevated during peak hours every day, and our autocorrelation method flags this link as congested during the periods shaded gray. The lower panel of Figure 3 shows the loss rate (computed over 5-minute intervals) during congested and uncongested intervals. We observe that a) loss

Far-end Higher During Congestion	Far-end Higher than Near-end	# Month-Links	% Month-Links
True	True	117	81%
True	False	12	8%
False	–	16	11%

Table 1: Correlation between congestion inferences and loss measurements for months-links that had a statistically significant difference in the far-end loss rate between congested and uncongested periods. For 145 such month-links, 129 had loss measurements consistent with the presence of congestion, and 117 of those passed the localization test.

rate to the far end was higher during congested periods than during uncongested periods and b) loss rate to the far end was higher than that to the near end during congested periods. Based on this intuition, we devised two statistical tests to analyze whether we can attribute an increased far-end loss rate to congestion on the interdomain link:

(1) *Far-end test*: is the far-end loss rate during congested periods *significantly higher* than during uncongested periods?
(2) *Localization test*: is the far-end loss rate during congested periods *significantly higher* than the near-end loss rate?

To apply these tests, we divided our data into *month-links*, each of which represents a month of data for one interdomain link probed from a single VP. Monthly blocks of data allow sufficient samples during congested and uncongested periods to distinguish loss rates. After filtering out month-links that were not significantly congested (at least one day during the month with at least 4% congestion, see §6), or where either the far or near end did not respond to our probes, we had 380 month-links, distributed across 162 interdomain links between 6 access providers and 31 transit providers. We further restricted our analysis to month-links that showed a statistically significant difference (irrespective of sign) in the far-end loss rate between congested and uncongested periods. We were left with 145 links after this filtering. We used the binomial proportion test (requiring $p < 0.05$) to evaluate these 145 month-links against the two tests, the results of which are summarized in Table 1.

Of these, 129 (89%, sum of first two rows) month-links passed the far-end test, i.e., loss rates significantly *increased* during congested periods. For 117 (81%, top row) month-links, the localization test also passed, i.e., far-end loss rate significantly exceeded near-end loss rate during periods of congestion; hence, we can attribute the increase in loss rate to the interdomain link. Of these 117 month-links, 5 had a suspiciously high loss rate (between 64–85%) to the far end at all times, suggesting a measurement artifact or ICMP rate limiting by the routers; however, the far-end loss rate still increased significantly during congested periods, and we therefore retain their classification in the top row of Table 1.

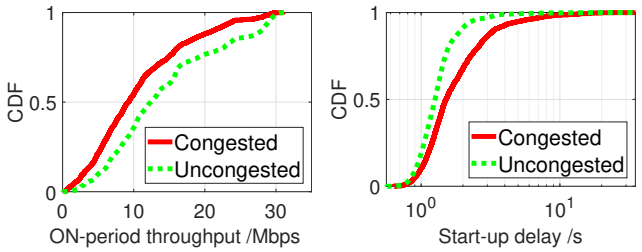
For 16 (11%, bottom row) of the 145 month-links in Table 1, the far-end loss rate decreased during congested periods as compared to uncongested periods. While these month-links seemingly contradict our hypothesis that congested periods would exhibit higher far-end loss rate, we find plausible explanations for 14 of them: episodes of high far-end loss uncorrelated with latency spikes, sometimes also with near-end loss (7 instances); high far-end loss (60–90%) at all times (3); large increases in the far-end loss in the last 3 days of the month with no elevated latency (2); high latency between near- and far-side routers suggesting an error in our border mapping (1); and insufficient data to infer congestion periods for most of the month (1).

5.2 Correlation with YouTube performance

We investigated how our inferences of interdomain congestion correlate with application performance metrics by studying YouTube video streaming, using both archived and new measurements taken with the `YouTube-test` tool [8]. We used 6 Ark VPs in the U.S. (the same VPs running NDT measurements) to run the YouTube performance measurements (§3.5) from October 2017 to the end of December 2017. To expand the scope of YouTube measurements, we used data from the same YouTube test deployed on 12 SamKnows VPs [14, 15] in the U.S. from May 2016 to July 2017.

We matched the interdomain links seen in traceroutes toward YouTube servers with those measured using TSLP per the *bdrmap*-based method used in [53], and selected interdomain links for which we had at least 50 YouTube streaming tests during periods that we inferred as congested using TSLP and the autocorrelation method. We identified a total of 17 congested interdomain links to Google from 7 SamKnows VPs connected to Comcast (16 links) and 1 Ark VP connected to CenturyLink (1 link).

We compared three streaming performance metrics during congested and uncongested periods. The first metric is ON-period throughput, which refers to the average instantaneous download rate of a video flow. After establishing a network connection to the YouTube server, a client starts filling its video buffer by downloading data using all available bandwidth. In steady state, traffic shows an ON-OFF pattern where during each ON-period, the client downloads a burst of packets from the video cache [40, 60]. We computed ON-period throughput as $T_{ON} = \frac{S_V}{\sum_{i=0}^{N} d_{ON}^i}$ where S_V is the total number of bytes of video data, N is the total number of ON periods in the steady state, d_{ON}^0 is the duration of



(a) ON-period throughput. **(b) Start-up delay (log).**

Figure 4: CDFs of YouTube streaming performance metrics in congested and uncongested periods.

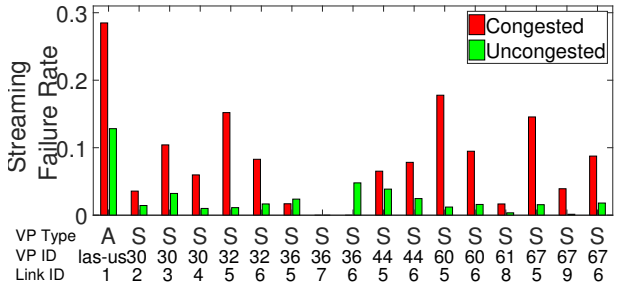


Figure 5: Median streaming failure rates for Ark (*A*) and SamKnows (*S*) VPs. The IDs of the VP and the interdomain link (labeled according to the far IP) are shown below the *x*-axis.

initial buffering phase, and $d^i_{ON} \forall i > 0$ is the duration of the i^{th} ON period. Figure 4a shows the CDFs of ON-period throughput measured across the selected interdomain links. During congested periods, the median throughput decreased 25.4% from 12.4 Mbps to 9.2 Mbps.

The second metric we considered is the startup delay, defined as the time to establish the connection and stream the first two seconds of video. The CDFs of the startup delay (Figure 4b) show that the median startup delay was inflated by 20.0% during congested periods. Further, only 67.9% of tests during congested periods (vs. 91.2% during uncongested periods) could start streaming within 2 seconds. This long delay could cause users to abandon the video [45].

Apart from degraded streaming performance, congestion could cause streaming failure, such as failing to download the next video segment during streaming. These failure events can manifest as rebuffering events on a viewer's player if the video buffer depleted before the YouTube video player could resume the download. Figure 5 shows that except for SamKnows VP 36, failure rates were generally higher during congested periods. For tests across link 5 from SamKnows VP 60, the failure rate was 13.7 times higher during congested periods than uncongested ones. For the Ark VP, almost 30% of the tests failed during congested periods.

In summary, we found that interdomain congestion inferred by TSLP correlated with lower ON-throughput, higher startup delay, and higher failure rate of YouTube tests, all of which provide validation of our congestion inferences.

5.3 Correlation with NDT throughput

As a third independent source of validation of the TSLP method, we ran a controlled experiment of the Network Diagnostic Test (NDT) from 6 Ark VPs from 15 Nov to 31 Dec, 2017. To select NDT test servers, we ran traceroutes from each VP toward a list of 104 U.S. M-lab NDT servers [3] in October 2017, used *bdrmap* inferences to identify interdomain links in the traceroutes, and TSLP data to infer congestion on these links (using the *level-shift* method, as the *autocorrelation* method was not operational at the time). If a congested interdomain link was on the path to multiple NDT servers, we chose the server closest to our VP in terms of RTT. We found seven M-lab NDT servers where a traceroute from the VP traversed links that showed some evidence of congestion in October 2017.

Of these seven cases, the *autocorrelation* method detected evidence of recurring congestion on three links during the NDT data collection period: Comcast-Tata (*Link 1*), Comcast-Tata (*Link 2*), and CenturyLink-Cogent (*Link 3*). For these links, we analyzed traceroutes from M-lab in November and December 2017, along with DNS names and *bdrmap* inferences to investigate path symmetry. We inferred that our NDT tests traversed *Link 1* (in New York) and *Link 3* (in Los Angeles) on both the forward and reverse path, but did not traverse *Link 2* (in Chicago) on the reverse path.

Table 2 shows NDT metrics during congested and uncongested periods for these 3 links, and also the p-value of the Student's *t*-test, indicating whether the difference between throughput in congested and uncongested periods was statistically significant. For *Link 1* and *Link 3*, periods that TSLP identified as congested correspond to lower throughput with statistical significance. The difference was particularly stark for *Link 1* (Comcast-Tata in New York), which showed a clear pattern of diurnal congestion (See Figure 6). Tests crossing *Link 3* (CenturyLink-Cogent in Los Angeles) showed a smaller (but statistically significant) difference between congested and uncongested throughput. *Link 3* showed evidence of congestion during only 21 of the 45 days of NDT data collection, and was only congested for 2.5% of the day (36 minutes) on average. This explains the small difference in throughput observed for this link. For *Link 2* (Comcast-Tata in Chicago), the difference in throughput during congested and uncongested periods was not statistically significant. As mentioned previously, the forward path from the Ark VP to the NDT server traversed *Link 2* in Chicago but the reverse path traversed an interdomain link between Comcast and

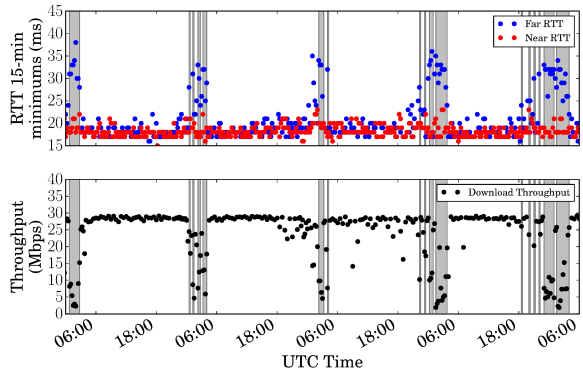


Figure 6: Time-series of TSLP latency and NDT throughput (Comcast-Tata, Link 1) on Dec 7–11, 2017. Periods identified as congested are shaded in gray.

Tata in Ashburn, VA that was not visible from any of our VPs. Our hypothesis is that the path from the NDT server to the Ark VP was not congested.

Link [VP AS - Server AS]	Uncong. Tput	Cong. Tput	*t*-test *p*-value
Link 1 [Comcast-Tata]	26.79	7.85	<0.001
Link 2 [Comcast-Tata]	23.75	23.55	0.324
Link 3 [CentLink-Cogent]	23.92	23.04	<0.001

Table 2: Average download throughput from NDT servers during congested and uncongested periods.

To summarize, drops in NDT throughput during periods we inferred as congested provided additional validation for our inferences. We note some caveats with correlating NDT and TSLP measurements. First, factors beyond interdomain link congestion could affect end-to-end throughput, e.g., home network congestion, server load, and congestion elsewhere on the end-to-end path [65]. For this reason, we do not use NDT throughput to infer interdomain link congestion. The second challenge is finding the right set of VPs and NDT servers to test congested links. Despite its large server footprint, M-lab covers a small fraction of interdomain links of large U.S. access ISPs [65], and running throughput measurements frequently can congest the VP's home network.

5.4 Operator feedback

We validated the inferences from our method directly with the operators of two large U.S. broadband access networks.

With the first operator, we shared our inferences for seven links to three transit providers and one content provider, and asked them if they observed recurring patterns of high link utilization. For six links we used data from one week in October 2017; for one link we observed recurring congestion in May 2017 that dissipated by October 2017, so we included data from both May and October. The operator confirmed that all our inferences were correct.

The second operator provided us confidential access to utilization data from their routers, which we used to validate our inferences on 20 links to two transit and two content providers in 2017. Of the 20 links, our method classified 10 as showing recurring congestion and 10 as uncongested at various times in 2017. In each case, the link utilization was consistent with our congestion inference for that link. Specifically, when our method flagged a link as experiencing recurring congestion, we observed the utilization of the link approach or reach 100% (true positives). For links where our method did not find evidence of congestion, the link utilization did not approach or reach 100% (true negatives).

6 U.S. INTERDOMAIN CONGESTION

Our probing system generates a large volume of data. Using Comcast (one of the largest ISPs in our dataset) as an example, and AS-relationships inferred by CAIDA's AS-relationship algorithm [50], we discovered links with 1353 customers, 108 peers, and 2 transit providers. We focused on measurements of interdomain links to peers and transit providers collected from March 2016 to December 2017, and for tractability limited our analysis to major peers and content providers. There were 34 ASes in this reduced set for Comcast. Further limiting our analysis to links we observed for at least seven days yielded a total of 973 links to those peers since March 2016. The population of links varies, as our visibility of interdomain links is dynamic. In December 2017, there were 345 visible links to this reduced set of peers.

The congestion numbers we report derive from our *autocorrelation* analysis method (§4.2). For each day for each link (which we refer to as a *day-link*), the algorithm classifies the day as *congested* or *uncongested*. If *congested*, it computes the duration of the congestion episode as a percentage of the day; a metric we call the *day-link congestion percentage*. We restrict our attention to links where the day-link congestion percentage was more than 4%, corresponding to approximately one hour of congestion per day. This restriction excluded from subsequent analysis 35.24% of the day-links that showed any congestion.

6.1 Overview of congested day-links

Table 3 provides a summary of the overall state of congestion during our 22-month observation window between U.S. access ISPs and the set of their interconnected ASes as defined above. It shows the number of peer/provider ASes (the reduced set selected as above) for each access ISP we probe,

Access Network	Obs. Peers & Providers	Cong. Peers & Providers	%Cong. Day-Links
CenturyLink	28	7	1.39
AT&T	34	7	2.58
Cox	20	5	8.41
Comcast	34	5	4.46
Charter	18	4	1.36
TWC	25	4	3.73
Verizon	26	3	3.09
RCN	19	1	0.52

Table 3: Observed (U.S.) transit and content providers, congested T&CPs, and % of congested day-links (to any T&CP) for each access network (Mar 2016 - Dec 2017).

the number of peer ASes that showed evidence of congestion during our measurement window, and the percentage of day-links that were congested. The numbers in Table 3 indicate that during our measurement period, congestion was not widespread on the peer/provider interdomain links we observed. Only a small fraction (between 5% and 25%) of peers/providers of each AP showed evidence of congestion at any time in our 22-month measurement study, and the overall fraction of congested day-links was also small — less than 5% for all ISPs except Cox (8.4%).

Table 4 provides a more detailed view of the data in Table 3, focusing on the transit and content providers that more commonly exhibited evidence of significant congestion. We computed the percentage of day-links showing evidence of congestion (*% congested day-links*) between each access provider (AP) and each transit/content provider (T&CP). To explore whether some T&CPs exhibited frequently congested day-links with many access providers, we ranked T&CPs based on the average % congested day-links to each of their connected APs. The T&CPs in Table 4 ranked at the top of that list, but these 9 T&CPs showed different congestion profiles to different APs. For instance, our system classified as congested 94% of the observed day-links between CenturyLink and Google, but only 1% of those between Cox and Google. Furthermore, APs exhibited different congestion profiles to different T&CPs: Comcast and Google had 22% of day-links classified as congested, whereas that number was only 1% for Comcast and Netflix. The prominence of Google in Table 4 is unsurprising, given their efforts to actively maintain high peering link utilization [71].

While no AP showed significant congestion to most of their interconnecting T&CPs, some showed significant congestion to specific T&CPs, particularly CenturyLink (to Google) and AT&T (to Tata). Finally, our system inferred as uncongested the vast majority (always over 90%) of observed day-links for each AP.

6.2 Temporal evolution of congestion

Tables 3 and 4 provide no insight into the dynamics of congestion over time. Figure 7 shows how congestion between specific APs and T&CPs evolved over an almost two-year period. Most patterns of congestion lasted less than six months, although some providers had a high percentage of day-links with congestion for prolonged periods, such as Comcast to Tata and NTT, or AT&T to Tata and XO. Other providers experienced congestion that dissipated later in the observation period. For example, congestion between Comcast and Google decreased from March 2016 to June 2016, peaked in December 2016, and dissipated in July 2017 until the end of our study. Interestingly, the dissipation of Comcast-Google congestion coincided with the rise of congestion on Comcast-Tata and Comcast-NTT links in the latter half of 2017. Patterns of rising and declining congestion were also evident for Verizon and AT&T to Google, and Cox to Netflix and Level3. TWC showed congestion to Tata, Vodafone, XO and Telia in 2016, all of which dissipated by December 2016.

6.3 Degree of congestion

Figure 7 summarizes the number of day-links that were congested more than 4% of the time. However, it masks an important question — how congested were those day-links? A day-link that is congested 5 or 10% of the time is less likely to contribute to degradation of the user experience than a day-link congested 50% or more. We define *mean congestion* between two networks over a month as the average percentage congestion on all day-links between those networks where any congestion was detected. Figure 8 plots this mean congestion metric for each month from each AP to two T&CPs — Google and Tata — that showed the most significant evidence of congestion. Even for the more frequently congested interdomain pairs seen in Figure 7, the mean congestion for each month was typically limited to a few hours per day (less than 5 hours, or 20%). However, we observed a high degree of variability between different pairs of APs and T&CPs. The links between Google and CenturyLink had a mean congestion percentage between 20–40% (approximately 5-10 hours per day) for 13 months, whereas other APs to Google were generally below 20%. Tata showed evidence of synchronized congestion upswings: in the latter half of 2016, multiple APs (AT&T, TWC, Comcast) experienced increases in mean congestion to Tata. In general, Tata exhibited mean congestion over 20% to one or more APs during the entire study period. As in Figure 7, we find trends over time in the mean congestion — notably for AT&T to Tata, which peaked in January 2017 and declined thereafter.

Transit/Access Provider	Comcast	Verizon	CenturyLink	AT&T	Cox	TWC	Charter	RCN
Google	21.63	25.47	94.09	15.05	1.36	–	3.01	Z
Tata	39.82	1.68	7.07	51.46	–	26.95	–	–
NTT	29.16	Z	Z	11.59	7.06	–	Z	Z
XO	6.33	0.35	5.25	15.27	–	8.17	4.82	–
Netflix	1.01	4.42	11.18	2.13	19.24	2.75	4.64	Z
Level3	1.29	0.63	3.69	3.80	32.28	1.81	Z	0.12
Vodafone	2.65	5.30	6.76	–	Z	2.09	–	–
Telia	2.37	0.90	0.60	11.89	Z	3.58	Z	Z
Zayo	0.34	0.11	0.39	Z	1.63	0.04	–	16.07

Table 4: Percentage of congested day-links for each pair of providers. These T&CPs represent 19% of U.S. networks studied, but represent 89% of *all* observed congested day-links. Z: congested day-links < 0.01%. –: No observations.

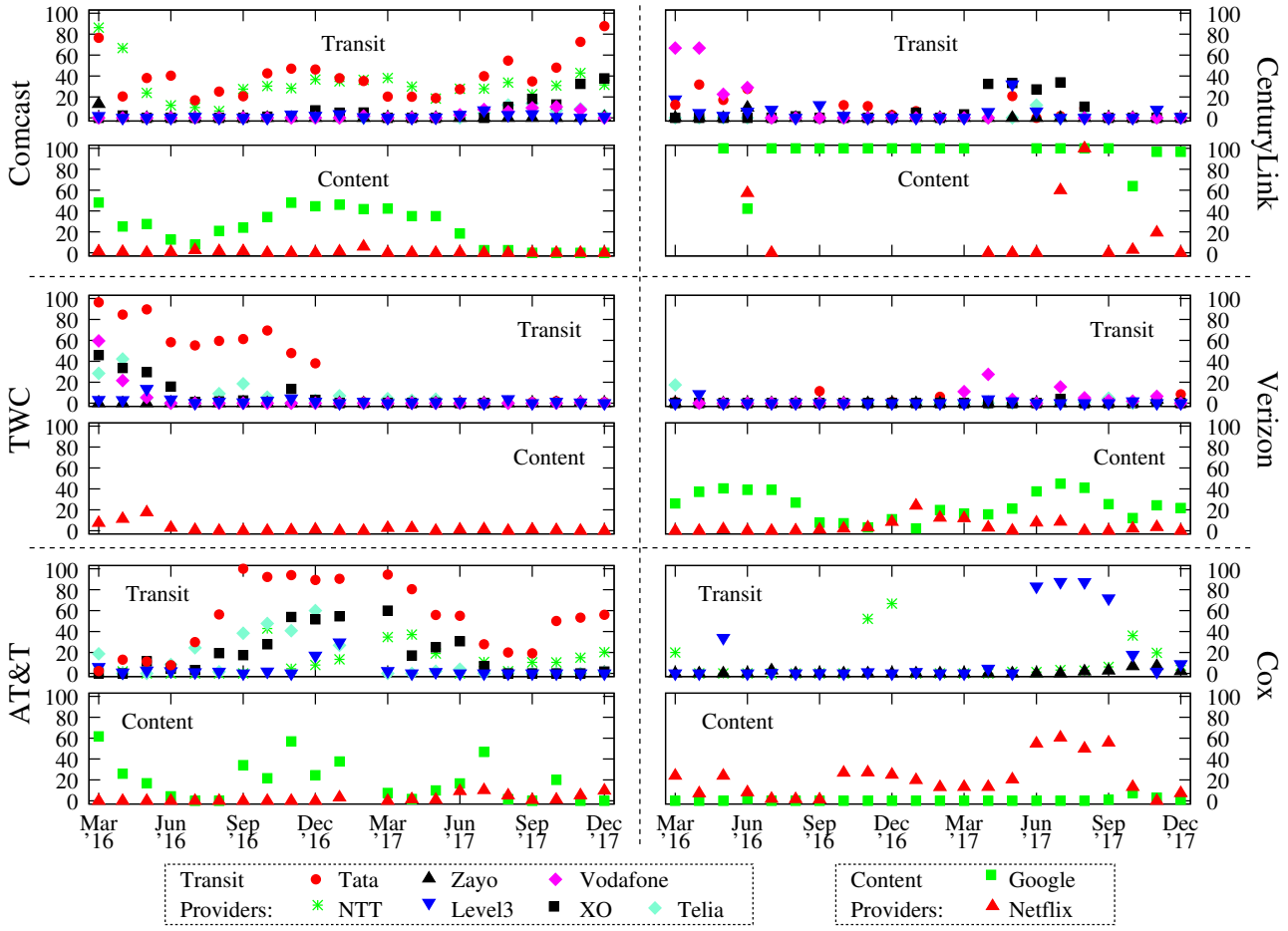


Figure 7: Percentage of day-links congested per month between an access provider and the frequently-congested T&CPs identified in Sec. 6.1. In most cases, congestion dissipates within a few months (typically 5 or less), but several access providers show congestion lasting 6 or more months to Google, Tata and NTT. Additionally, a single access provider has congestion lasting 6 months or more to Netflix and XO.

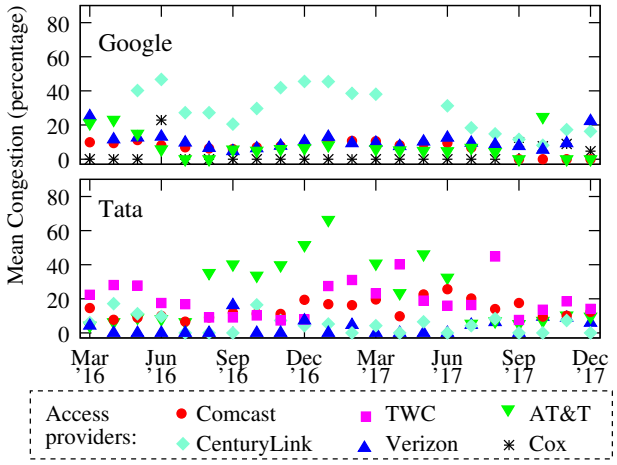


Figure 8: Mean day-link congestion percentage for the two most frequently-congested T&CPs to all measured APs. Google is similarly congested to all APs for most of the measurement period, with the notable exception of CenturyLink. In general, Tata exhibits higher mean congestion to more APs than Google.

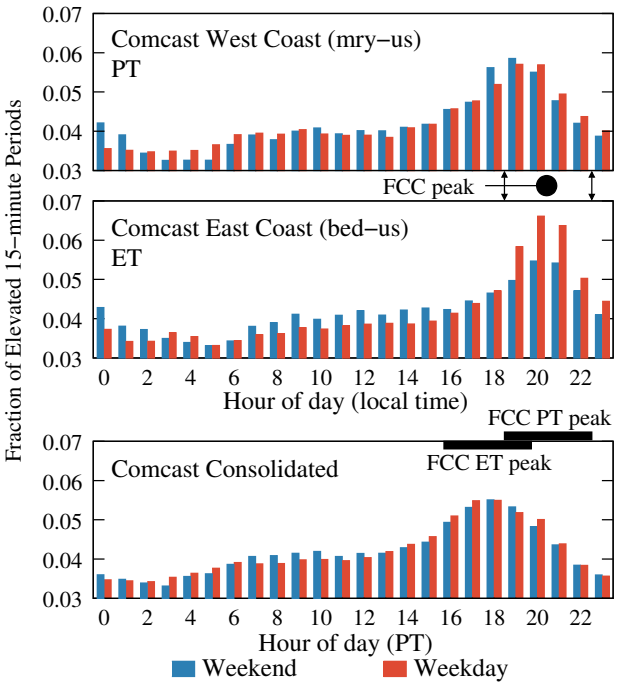


Figure 9: Distribution of recurring 15-minute congestion periods from 2017 as seen from VPs in Comcast.

6.4 Case study of Comcast

We conclude with a case study highlighting time-of-day and day-of-week effects in congestion inferred on peers/providers of Comcast. Figure 9 shows the hourly distribution of 15-minute periods from 2017 inferred as showing recurring congestion according to our autocorrelation method. The top two histograms plot the fraction of elevated 15-minute periods that fall in each hourly bin for all links measured from two VPs (one on the East coast and one on the West coast of the U.S.) using local time at the VP. The bottom histogram plots the same fraction aggregated for all links measured from all VPs in Comcast. While our congestion inference methods do not use time of day information in making inferences, the autocorrelation-generated congestion periods occurred during *peak hours* as defined by the F.C.C.'s Measuring Broadband America program [34] (7pm to 11pm local time). Note that the mode of the pdf for the East coast VP is at 8pm local time while that for the West coast is 7pm local time. It is tempting to speculate about whether the East and West coasts experience peak Internet traffic demands at different times, but a further effect is in play here: each VP measures interdomain links in other time zones as well as its own. Without access to accurate router geolocation data, we defer an analysis of this phenomenon to future work. The figure also shows that weekends have similar congestion patterns as weekdays, in contrast to the FCC's classification of weekends as off-peak periods.[1]

7 LIMITATIONS

We are aware of several limitations of our methodology.

Router Queueing Behavior: Our method relies on routers queueing ICMP probe packets similarly to regular traffic. A router that de-prioritizes ICMP responses or generates them in the slow-path could inflate observed latencies. In practice we have found that such latency patterns lack an identifiable diurnal pattern, and hence are not classified as congestion. Another possibility is that ISPs game our technique by *prioritizing* ICMP responses, which could cause our technique to mis-identify congested links as non-congested.

Incompleteness: Our current system does not provide visibility of all interdomain links of U.S. access providers, and the links we measured may not be representative of all such links. Observing all interdomain links requires numerous VPs in geographically diverse locations within the same network [49], which Ark does not provide. We plan to address this limitation by expanding our set of VPs (§9).

Root causes unknown: Determining the *root cause* of a congested interdomain link, or attributing responsibility to a specific network is not possible with the data we collect. Congestion could be due to a peering dispute, due to one network operating the link at high utilization [71], or due to traffic engineering by networks sourcing traffic.

[1]The technical appendix of [34] states that "At peak hours, defined for this study as the period on weekdays between 7:00 pm and 11:00 pm local time.."

Link capacity information: Our methods do not measure link utilization or capacity. We give all links between two networks equal weight, although they may have different capacities, and hence different impact when congested.

Asymmetric routes: Asymmetric routes (§4.2) may cause responses to our probes to traverse a different link than the targeted interdomain link. In the case of hot-potato routing, our probes will normally follow a symmetric path back to the VP. While the path to the ultimate destination may be asymmetric, for a probe that terminates at the far end of an interconnection, the closest path back to the VP is across that same link. The case where we see evidence that our probes follow an asymmetric path is a neighbor network that delivers packets (including responses to our probes) to the VP using the interconnection point that is closest to the VP. Our initial exploration of this case suggests it is rare. We have several potential techniques to detect these cases, including identifying significant differences in baseline delays to the near and far sides of the link, and use of the IP record route option. Another approach to determine the return path relies on extracting a long-term *congestion signature* of the path from our data. We have found that a simple correlation between two TSLP time-series provides a good indication that return traffic from those two targets traversed the same congested path. We plan to further explore means to detect and characterize asymmetric paths.

8 POLICY IMPLICATIONS

Our original motivation for this work was an increase in heated peering disputes between powerful players in the U.S. which raised questions about intentional degradation of performance as a business strategy to obtain (or avoid) interconnection fees [70]. The prevalence of these public disputes dropped around the time of the FCC's 2015 Open Internet Order, in which the FCC asserted authority over interconnection, sending a signal to industry to resolve disputes or trigger regulatory oversight.[2] However, our measurements reveal indications of persistently congested transit links, which – regardless of cause – implies clear motivation for large players to engage in direct peering negotiations.

The FCC recognized that they lacked sufficient understanding of interconnection to impose any regulations [37]. In part to close this gap in understanding, during the next merger between an access and content provider (AT&T and DirecTV [36]), the FCC imposed interconnection measurement and reporting conditions, for 4 years, under NDA agreements [26]. Like other sources of interconnection data, this data tells a partial story, but in this case, a secret one [44].

[2]We cannot prove causation there; commercial players may have realized independent motivations to resolve the peering disputes.

Thus, the most important contribution of this work is addressing this decades-long gap in an objective third-party's ability to study peering disputes in an open, objective, scientifically validated way. Especially in today's deregulatory political climate, we consider such measurement to be the most promising strategy for incentivizing good ISP behavior.

9 FUTURE DIRECTIONS

We will support community use of our raw data and tools, and solicit feedback on our web interface and API access to the raw congestion measurements. We hope to eventually provide an interface where interested parties can make more complicated queries against this and related network-level topology data, hopefully leveraging infrastructure geolocation capabilities (albeit sparse).

We hope to address the incompleteness in our coverage by expanding our VPs to use the FCC's Measuring Broadband America (MBA) infrastructure [38] consisting of thousands of home routers. Improving the quality of data, as well as expanding to thousands of additional FCC MBA VPs will bring new challenges in system scale and data processing.

Our current system limits congestion measurements to interdomain links identified by *bdrmap*, which only identifies immediate neighbors of the network hosting a measurement VP. Recently, Marder et al. [52] introduced the MAP-IT tool to identify interdomain links in a set of collected traceroutes. We will investigate whether the combination of *bdrmap* and *MAP-IT* can enable measurement of interdomain links farther than one AS hop away from the network hosting our VP.

As we demonstrated in Section 6, interdomain congestion shows temporal effects over timescales of weeks and months. Such effects could be the result of complex traffic engineering and capacity augmentation by the involved networks. In future work we will investigate whether we can correlate temporal variations in congestion with routing and topological changes observable in BGP.

Finally, while we demonstrated that the TSLP technique can identify congested links, the impact of congestion on user QoE remains a largely open question. In our ongoing work, we are building a system to crowd-source QoE measurements such as video streaming performance and webpage loads (similar to the Eyeorg platform [68]), and correlate those with congestion inferences from TSLP.

ACKNOWLEDGMENTS

We thank our shepherd Dina Papagiannaki and the anonymous reviewers for their insightful comments, and Ahmed Elmokashfi for providing an implementation of the level-shift algorithm. This work was partly funded by NSF grants CNS-1413905 and CNS-1414177.

REFERENCES

[1] 2017. Grafana: The Open Platform for Beautiful Analytics and Monitoring. https://grafana.com/.

[2] 2017. InfluxDB. https://www.influxdata.com/.

[3] 2017. NDT (Network Diagnostic Tool) - M-Lab. https://www.measurementlab.net/tests/ndt/.

[4] 2017. PeeringDB. http://www.peeringdb.com.

[5] 2017. RIPE RIS. http://www.ripe.net/ris/.

[6] AfriNIC. 2017. AfriNIC delegaged files. ftp://ftp.afrinic.net/.

[7] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. 2012. Anatomy of a Large European IXP. In *Proceedings of ACM SIGCOMM*.

[8] Saba Ahsan, Vaibhav Bajpai, Jörg Ott, and Jürgen Schönwälder. 2015. Measuring YouTube from Dual-Stacked Hosts. In *Proceedings of the Passive and Active Measurement Conference (PAM)*.

[9] Collin Anderson. 2015. New Opportunities for Test Deployment and Continued Analysis of Interconnection Performance. http://www.measurementlab.net/blog/interconnection$_a$nd$_m$easurement$_u$pdate.

[10] Robert Andrews and Stacey Higginbotham. 2013. YouTube Sucks on French ISP Free, and French Regulators Want to Know Why. http://gigaom.com/2013/01/02/youtube-sucks-on-french-isp-free-french-regulators-want-to-know-why/.

[11] APNIC. 2017. APNIC delegated files. ftp://ftp.apnic.net/pub/stats/apnic/.

[12] ARIN. 2017. ARIN delegated files. ftp://ftp.arin.net/pub/stats/arin/.

[13] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. 2006. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[14] Vaibhav Bajpai, Saba Ahsan, Jürgen Schönwälder, and Jörg Ott. 2017. Measuring YouTube Content Delivery over IPv6. *ACM SIGCOMM Computer Communication Review* 47, 5 (Oct. 2017), 2–11.

[15] V. Bajpai and J. Schönwälder. 2015. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *IEEE Communications Surveys Tutorials* 17, 3 (2015), 1313–1341.

[16] Steven Bauer, David Clark, and William Lehr. 2009. The Evolution of Internet Congestion. In *Research Conference on Communication, Information and Internet Policy (TPRC)*.

[17] Jon Brodkin. 2013. Time Warner, Net Neutrality Foes Cry Foul Over Netflix Super HD Demands. http://arstechnica.com/business/2013/01/timewarner-net-neutrality-foes-cry-foul-netflix-requirements-for-super-hd/.

[18] Jon Brodkin. 2013. Why YouTube Buffers: The Secret Deals that Make-and-break Online Video. http://arstechnica.com/information-technology/2013/07/why-youtube-buffers-the-secret-deals-that-make-and-break-online-video/.

[19] Sean Buckley. 2013. France Telecom and Google Entangled in Peering Fight. http://www.fiercetelecom.com/story/france-telecom-and-google-entangled-peering-fight/2013-01-07.

[20] CAIDA. 2017. AS Relationships. http://www.caida.org/data/as-relationships/.

[21] CAIDA. 2017. Inferred AS to Organization Mapping Dataset. https://www.caida.org/data/as-organizations/.

[22] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. 2013. Mapping the Expansion of Google's Serving Infrastructure. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[23] Balakrishnan Chandrasekaran, Georgios Smaragdakis, Arthur Berger, Matthew Luckie, and Keung-Chi Ng. 2015. A Server-to-server View of the Internet. In *Proceedings of ACM CoNEXT*.

[24] Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, and Walter Willinger. 2004. Towards Capturing Representative AS-level Internet Topologies. *Computer Networks* 44, 6 (2004), 737–755.

[25] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. 2015. Are We One Hop Away from a Better Internet?. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[26] KC Claffy, Amogh Dhamdhere, David Clark, and Steven Bauer. 2016. First Amended Report of AT&T Independent Measurement Expert: Reporting requirements and measurement methods. https://www.caida.org/publications/papers/2016/att$_i$me$_f$irst$_a$mended$_r$eport/.

[27] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. 2002. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proceedings of ACM SIGCOMM*.

[28] Leiwen Deng and Aleksandar Kuzmanovic. 2008. Monitoring Persistently Congested Internet Links. In *Proceedings of the International Conference on Network Protocols (ICNP)*.

[29] Amogh Dhamdhere and Constantine Dovrolis. 2010. The Internet is Flat: Modeling the Transition From a Transit Hierarchy to a Peering Mesh. In *Proceedings of ACM CoNEXT*.

[30] Florin Dinu and T. S Eugene Ng. 2011. Inferring a Network Congestion Map with Zero Traffic Overhead. *Proceedings of the International Conference on Network Protocols (ICNP)*.

[31] Joan Engebretson. 2010. Level 3/Comcast Dispute Revives Eyeball vs. Content Debate. http://www.telecompetitor.com/level-3comcast-dispute-revives-eyeball-vs-content-debate.

[32] Joan Engebretson. 2013. Behind the Level 3-Comcast Peering Settlement. http://www.telecompetitor.com/behind-the-level-3-comcast-peering-settlement/.

[33] R. Fanou, F. Valera, and A. Dhamdhere. 2017. Investigating the Causes of Congestion on the African IXP substrate. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[34] FCC's Office of Engineering and Technology and And Consumer and Governmental Affairs. 2016. Measuring Broadband America Fixed Broadband Report. http://data.fcc.gov/download/measuring-broadband-america/2016/2016-Fixed-Measuring-Broadband-America-Report.pdf.

[35] Nick Feamster. 2016. Revealing Utilization at Internet Interconnection Points. In *Research Conference on Communication, Information and Internet Policy (TPRC)*.

[36] Federal Communications Commission. 2015. In the Matter of Applications of AT&T Inc. and DIRECTV For Consent to Assign or Transfer Control of Licenses and Authorizations: MB Docket No. 14-90. https://apps.fcc.gov/edocs$_p$ublic/attachmatch/FCC-15-94A1.pdf.

[37] Federal Communications Commission. 2015. In the Matter of Protecting and Promoting the Open Internet: Report and Order on Remand, Declaratory Ruling, and Order: GN Docket No. 14-28. https://apps.fcc.gov/edocs$_p$ublic/attachmatch/FCC-15-24A1.pdf.

[38] Federal Communications Commission. 2017. Measuring Broadband America. https://www.fcc.gov/general/measuring-broadband-america.

[39] Romain Fontugne, Cristel Pelsser, Emile Aben, and Randy Bush. 2017. Pinpointing Delay and Forwarding Anomalies Using Large-scale Traceroute Measurements. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[40] Monia Ghobadi, Yuchung Cheng, Ankur Jain, and Matt Mathis. 2012. Trickle: Rate Limiting YouTube Video Streaming. In *Proceedings of USENIX ATC*.

[41] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. 2008. The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse?. In *Proceedings of the Passive and Active Network Measurement Conference (PAM)*.

[42] Google Video Quality Report. 2016. https://www.google.com/get/videoqualityreport/.

[43] Peter J. Huber. 1981. *Robust Statistics*. Wiley.

[44] kc Claffy, David D Clark, Steve Bauer, and Amogh Dhamdhere. 2016. Policy Challenges in Mapping Internet Interdomain Congestion. In *Research Conference on Communication, Information and Internet Policy (TPRC)*.

[45] S. Shunmuga Krishnan and Ramesh K. Sitaraman. 2013. Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs. *IEEE/ACM Transactions on Networking* 21, 6 (Dec. 2013), 2001–2014.

[46] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. 2010. Internet Inter-Domain Traffic. In *Proceedings of ACM SIGCOMM*.

[47] LACNIC. 2017. LACNIC delegated files. ftp://ftp.lacnic.net/pub/stats/lacnic/.

[48] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, and k. claffy. 2014. Challenges in Inferring Internet Interdomain Congestion. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[49] M. Luckie, A. Dhamdhere, B. Huffaker, D. Clark, and k. claffy. 2016. bdrmap: Inference of Borders Between IP Networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[50] M. Luckie, B. Huffaker, k. claffy, A. Dhamdhere, and V. Giotsas. 2013. AS Relationships, Customer Cones, and Validation. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[51] M-Lab Research Team. 2014. ISP Interconnection and its Impact on Consumer Internet Performance - A Measurement Lab Consortium Technical Report. http://www.measurementlab.net/publications.

[52] Alexander Marder and Jonathan M. Smith. 2016. MAP-IT: Multipass Accurate Passive Inferences from Traceroute. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[53] Ricky Mok, Vaibhav Bajpai, Amogh Dhamdhere, and kc Claffy. 2018. Revealing the Load Balancing Behavior of YouTube Traffic on Interdomain Links. In *Proceedings of the Passive and Active Measurement Conference (PAM)*.

[54] Netflix. 2017. ISP Speed Index. https://ispspeedindex.netflix.com/country/us/.

[55] University of Oregon. 2017. RouteViews. http://www.routeviews.org/.

[56] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2010. The (In)completeness of the Observed Internet AS-level structure. *IEEE/ACM Transactions on Networking* 18, 1 (2010), 109–122.

[57] Ricardo V. Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2008. In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. In *Proceedings of ACM SIGMETRICS*.

[58] Ricardo V. Oliveira, Beichuan Zhang, and Lixia Zhang. 2007. Observing the Evolution of Internet AS Topology. In *Proceedings of ACM SIGCOMM*.

[59] Packet Clearing House. 2017. Full Exchange Point Dataset. https://prefix.pch.net/applications/ixpdir/menu$_d$ownload.php.

[60] Ashwin Rao, Yeon sup Lim, Chadi Barakat, Arnaud Legout, Don Towsley, and Walid Dabbous. 2011. Network Characteristics of Video Streaming Traffic. In *Proceedings of ACM CoNEXT*.

[61] RIPE NCC. 2017. RIPE NCC delegated files. ftp://ftp.ripe.net/ripe/stats/.

[62] Matthew Roughan, Jonathan Tuke, and Olaf Maennel. 2008. Bigfoot, Sasquatch, the Yeti and Other Missing Links: What we Don't Know About the AS Graph. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[63] Sandvine. 2013. Global Internet Phenomenon Report, 1H13.

[64] S. Sundaresan, A. Dhamdhere, M. Allman, and k. claffy. 2017. TCP Congestion Signatures. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[65] S. Sundaresan, D. Lee, X. Deng, Y. Feng, and A. Dhamdhere. 2017. Challenges in Inferring Internet Congestion Using Throughput Measurements. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*.

[66] Mark Taylor. 2014. Observations of an Internet Middleman. http://blog.level3.com/global-connectivity/observations-internet-middleman/.

[67] W. A. Taylor. 2000. Change-point Analysis: A Powerful New Tool for Detecting Changes. http://www.variation.com/cpa/tech/changepoint.html.

[68] Matteo Varvello, Jeremy Blackburn, David Naylor, and Konstantina Papagiannaki. 2016. EYEORG: A Platform For Crowdsourcing Web Quality Of Experience Measurements. In *Proceedings of ACM CoNEXT*.

[69] Verizon. 2013. Unbalanced Peering, and the Real Story Behind the Verizon/Cogent Dispute. http://publicpolicy.verizon.com/blog/entry/unbalanced-peering-and-the-real-story-behind-the-verizon-cogent-dispute.

[70] William Norton. 2010. The Art of Peering: The Peering Playbook. http://drpeering.net/white-papers/Art-Of-Peering-The-Peering-Playbook.html.

[71] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, Victor Lin, Colin Rice, Brian Rogan, Arjun Singh, Bert Tanaka, Manish Verma, Puneet Sood, Mukarram Tariq, Matt Tierney, Dzevad Trumic, Vytautas Valancius, Calvin Ying, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *Proceedings of ACM SIGCOMM*.