



Advancing the Art of Internet Edge Outage Detection

Philipp Richter
MIT / Akamai
richterp@csail.mit.edu

Ramakrishna Padmanabhan
University of Maryland
ramapad@cs.umd.edu

Neil Spring
University of Maryland
nspring@cs.umd.edu

Arthur Berger
Akamai / MIT
awberger@akamai.com

David Clark
MIT
ddc@csail.mit.edu

ABSTRACT

Measuring reliability of edge networks in the Internet is difficult due to the size and heterogeneity of networks, the rarity of outages, and the difficulty of finding vantage points that can accurately capture such events at scale. In this paper, we use logs from a major CDN, detailing hourly request counts from address blocks. We discovered that in many edge address blocks, devices, collectively, contact the CDN every hour over weeks and months. We establish that a sudden temporary absence of these requests indicates a loss of Internet connectivity of those address blocks, events we call *disruptions*.

We develop a disruption detection technique and present broad and detailed statistics on 1.5M disruption events over the course of a year. Our approach reveals that disruptions do not necessarily reflect actual *service outages*, but can be the result of prefix migrations. Major natural disasters are clearly represented in our data as expected; however, a large share of detected disruptions correlate well with planned human intervention during scheduled maintenance intervals, and are thus unlikely to be caused by external factors. Cross-evaluating our results we find that current state-of-the-art active outage detection over-estimates the occurrence of disruptions in some address blocks. Our observations of disruptions, service outages, and different causes for such events yield implications for the design of outage detection systems, as well as for policymakers seeking to establish reporting requirements for Internet services.

CCS CONCEPTS

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

KEYWORDS

Internet reliability, Internet outages

ACM Reference Format:

Philipp Richter, Ramakrishna Padmanabhan, Neil Spring, Arthur Berger, and David Clark. 2018. Advancing the Art of Internet Edge Outage Detection. In *2018 Internet Measurement Conference (IMC '18)*, October 31–November 2, 2018, Boston, MA, USA. ACM, New York, NY, USA, Article 4, 14 pages. <https://doi.org/10.1145/3278532.3278563>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IMC '18, October 31–November 2, 2018, Boston, MA, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5619-0/18/10...\$15.00

<https://doi.org/10.1145/3278532.3278563>

1 INTRODUCTION

Residential Internet access has become critical: while long ago packets were exchanged over a connection designed for reliable voice communication, it has become common to provide myriad services, even voice, over a network built for best-effort data communication. Reliable Internet connectivity has become increasingly necessary not only for individual users and their emergency communication needs or the operation of smart in-home devices but also for businesses that provide network services or sell to end-users. The increasing criticality of the Internet is reflected in growing attention from governments and regulators around the globe to monitor and improve Internet reliability [38–41].

Despite the importance of continuous Internet access, there is a shortage of high quality data that quantifies Internet reliability at the edge. Reliability is difficult to measure, since true outages are rare events happening inside of thousands of independently administered networks. Each network is subject to individual network management practices, resulting in different network characteristics, making it both challenging to develop methods to measure reliability at scale, as well as to interpret measurement results. While these challenges apply to reliability measurements of both residential and core networks, of particular relevance to end-user network outages is the typical absence of any global routing-protocol activity (§ 7). Prior approaches to measure Internet edge reliability have thus either relied on deploying hardware in end-user premises [27, 52, 55], or on periodically sending probe traffic to specific IP addresses [25, 46, 54].

This paper introduces a new passive approach to detect Internet edge disruptions and outages, using server logs of one of the world's largest CDNs. This dataset has several advantages over prior techniques: it samples the edge of the Internet broadly, it favors actively used addresses, and it relies on network traffic that is unlikely to be blocked. We make the following high-level contributions:

- **Measurement methodology:** We introduce a passive approach to detect disruptions in address activity based on CDN access logs. We leverage a key observation that an ever-increasing number of always-on devices (e.g., smartphones, smart TVs) result in constant, non-human triggered request activity to CDN servers, an effect we call *baseline activity*. We establish that a temporary absence of these requests indicates a loss of Internet connectivity of given address blocks, events we refer to as *disruptions*. We develop and evaluate an approach to robustly detect such disruptions in our dataset, enabling us to continuously track disruptions (i) on a broad scale, and (ii) in detail, i.e., for individual IPv4 /24 address blocks. Evaluating our approach

against Trinocular [46], a state-of-the-art Internet-wide active outage detection technique, we find that Trinocular’s outages must be filtered for most events to be correct, otherwise false positives in few address blocks can dominate.

- **Fine-grained understanding of disruptions:** We present detailed statistics on some 1.5M detected disruption events spanning one entire year. We investigate disruption sizes, duration, frequency of occurrence, and timing. While major external events such as natural disasters are clearly represented in our datasets, we find that a large share of disruptions are unlikely to be caused by external factors, but align well with scheduled ISP maintenance intervals. We illustrate our findings with a case study of major US broadband ISPs.
- **Disruptions vs. service outages:** We leverage an orthogonal dataset that enables us to track the activity of individual devices across address blocks in the face of disruptions. Our analysis reveals that at least some 10% of disruption events do not reflect actual service outages, but large-scale prefix migration. We discover that temporary prefix migrations often result in massive anti-disruption events, sudden shifts in prefix activity. We develop techniques to detect anti-disruptions on a per-AS level, and pinpoint networks that are particularly prone to show such behavior (and thus bias outage detection mechanisms). We study to what extent publicly available BGP data captures detected disruptions, finding that BGP hides some 80% of identified disruptions, but also that even a BGP withdrawal of a prefix does not necessarily indicate an actual service outage.

Our findings challenge common assumptions in the field of Internet edge outage detection, such as how to determine if a measured event really corresponds to a service outage. As well, our findings challenge the interpretation of such results, given that disruptions and outages can be caused by a variety of factors, i.e., whether a planned service maintenance should be interpreted similarly to a service outage caused by unplanned internal or external events.

The remainder of this paper is structured as follows: We introduce definitions and discuss related work in Section 2. In Section 3 we show how we can leverage baseline activity in the CDN logs to identify disruptions, introduce our detection mechanism and compare it against state-of-the-art active outage detection. We study identified disruption events on a broad scale in Section 4. We then shift our perspective and drill into details of disruption events from a device-centric perspective in Section 5. We discover and analyze the phenomenon of anti-disruptions in Section 6 and assess ways to distinguish disruptions from service outages in Section 7. We illustrate our findings with a case study of major US ISPs in Section 8 and discuss the pertinent implications of our work in Section 9.

2 ON DETECTING EDGE OUTAGES

In this section, we introduce necessary terminology and discuss the current state-of-the-art in Internet edge outage detection.

2.1 Defining Outages

In this work, we introduce a rigorous distinction between a detectable symptom of a service outage and the outage itself; and we consider possible, alternative causes of that symptom. We introduce the following two terms:

Disruption: A temporary loss of Internet connectivity of specific IP address blocks.

Outage: A disruption that results in the loss of the Internet access service that had been provided to the end devices in the affected address blocks.

A disruption may be the measurable consequence of an outage, but a disruption does not always imply that an actual outage occurred. For example, a disruption occurs when the public IP addresses associated with end hosts are changed and the prior addresses are not immediately assigned to other devices—an outage need not have occurred. There are different datasets and methods to detect disruptions in the Internet in the control and data planes, including the measurement of BGP announcements and withdrawals, sudden loss of ICMP responsiveness for specific address blocks, and sudden drops in traffic from/to specific address blocks.

2.2 Related Work

Internet failures affecting the core of the network have been well studied, using data-plane techniques [13, 43], control-plane techniques [26, 35], or combinations of both [23, 32–34]. Other works investigated external sources such as router logs [59] and mailing lists [11] to study infrastructure outages.

Failures affecting the edge have been studied at smaller scales using measurement agents deployed at user premises [10, 15, 16, 27, 55]. These agents are typically dedicated hardware devices, such as SamKnows [52] and BISmark [58] routers and RIPE Atlas probes [50], although some approaches use measurements from software deployed on user systems [10, 53, 56] or a combination of hardware and software [16]. Such approaches can offer detailed and accurate reports about Internet reliability since the agents are designed to execute measurements continuously as long as they are powered. However, cost and logistical difficulties of deploying measurement agents to users severely limits their scalability.

To detect outages at scale, studies have investigated actively probing destinations from vantage points and using probe responses—or lack thereof—as signal for edge outages. Thunderping analyzes the effect of weather on residential networks [54] by pinging residential IP addresses in geographic areas subject to severe weather. Trinocular [46] models the responsiveness of routed /24 prefixes using historical data [28] and sends ICMP probes to 4M routed /24 prefixes to detect disruptions. By applying Bayesian inference to responses, Trinocular detects a disruption affecting a prefix when it finds that the prefix has become unresponsive according to its model. We evaluate our results against Trinocular in § 3.7.

Dainotti et al. detect Internet outages at the country level by identifying times of reduced traffic from addresses in certain countries toward unused IPv4 address space [22]. Traffic to unused portions of the IPv4 address space is often sent by misconfigured devices or malicious hosts [12] who may spoof their source addresses, making it difficult to infer if addresses sending traffic to the darknet are actively in use by user devices.

Prior work has interpreted intermittently inactive addresses or address blocks (i.e., what we term *disruptions*) as outages [22, 46, 54]. A key aspect that differentiates our work is that we seek to further investigate whether detected disruptions result in service outages.

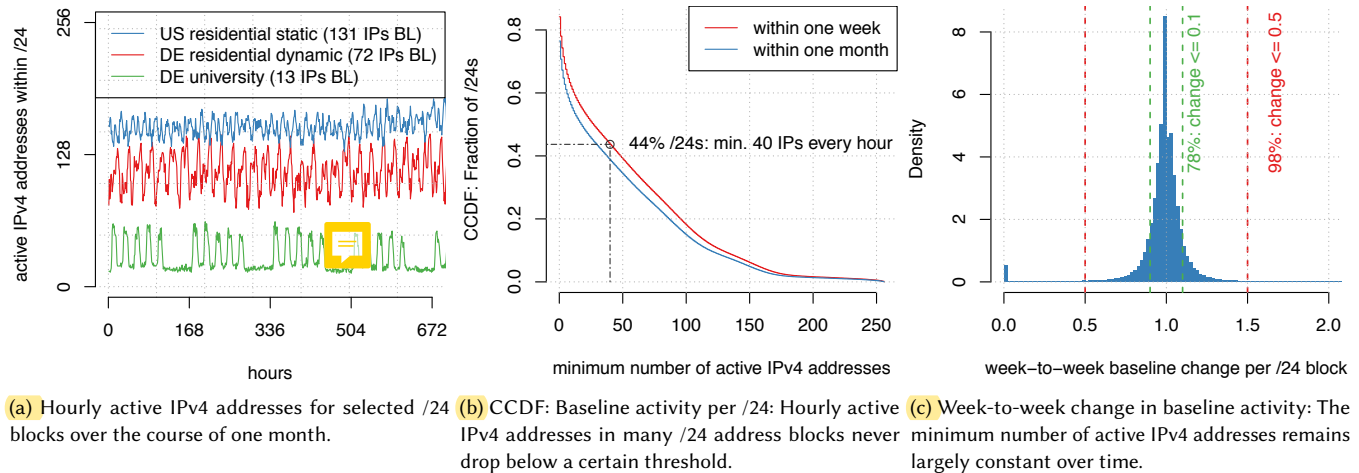


Figure 1: Baseline Activity: Minimum Number of active IPv4 addresses.

3 DETECTING DISRUPTIONS

In this section we first introduce our dataset and its properties. We discover and study the notion of *baseline* activity in our data and introduce our approach to detect disruptions. We then tune our parameters for robustness by cross-validating against **ICMP-based disruption detection** and compare our results with a state-of-the-art active outage detection mechanism.

3.1 Dataset

We base this study on (proprietary) server logs of one of the world's largest CDNs which operated more than 240,000 servers in more than 130 countries and over 1,700 networks, serving some 3 trillion HTTP requests on a daily basis. Each time a client fetches a Web object from a CDN edge server, a log entry is created, which is then processed and aggregated through a distributed data collection framework. Our dataset consists of the number of requests ("hits") per hour issued by each IP address over an observation period of 54 weeks from March 2017 to March 2018. We note that due to the hourly **binning** of our dataset, we can only detect disruptions that span at least *one full calendar hour*. Prior work established that the CDN logs capture activity from the vast majority of the active address space (some 1.2B active IPv4 addresses over the course of a year), and we refer the reader to [48] for a more detailed study of the visibility of the CDN logs.

The CDN's servers are typically located close to the end-users, often in the same network, and/or metropolitan area. Also, if there is a fault in the path between the client device and the server, or the server goes down, or there is a problem in the datacenter hosting the server, but the client still has connectivity to the Internet, then connections to the CDN may be disrupted, but can be re-established to another server, possibly in a different datacenter. The CDN continuously revises the DNS forward resolution, and the DNS TTLs are nominally 20 seconds. Thus, over a period of an hour, a drop in the number of hits from a given address is due to either the associated end devices not attempting to connect to the CDN, or the loss of connectivity at that address, possibly due to planned operations of the network provider, or an unplanned event.

3.2 Steady CDN Activity as Signal

The hourly snapshots from our logs provide a time series to analyze. As expected, hourly hit counts (traffic) have both diurnal and day-of-the-week effects, as well as other effects, such as holidays and other variations in activity from end devices. There is a large literature on detecting anomalies in time series (e.g., [9, 20, 31, 57, 60]), and we tried various methods. However, we soon realized that we then faced the difficult problem of determining which detected anomalies in the time series were actually a disruption, i.e., **loss of Internet connectivity of the address blocks**.

Instead, we selected a subset of the prefixes for which we can infer an *activity signal that is both largely independent of direct human-triggered activity and is dependent on a functioning network*. We find that the number of addresses active in a given hour yields a smoothed signal of the number of requests per hour, and that for many prefixes, this number of active addresses has a high-enough baseline (minimum over a week-long interval) to permit observing a disruption as a significant violation of this minimum. We focus on this *baseline address activity* metric and apply our technique only to those prefixes that have a sustained, sufficiently high baseline. We next show examples of baseline address activity, how prevalent a high baseline is, and how stable it is.

Baseline address activity examples: Figure 1a shows the number of hourly active IPv4 addresses from three selected /24 address blocks over one month. Although individual address blocks vary widely in terms of active addresses, note that each shows a *baseline activity*, i.e., the number of addresses contacting the CDN has a relatively stable minimum value. We manually inspected off-hour request traffic from several address blocks and found that a variety of Smartphone applications, widgets, and software installations cause this activity by sending repeated beacons, status updates, and update requests. Thus, baseline activity persists at any given hour and does not require action by humans, e.g., by visiting a website hosted on the CDN's infrastructure.

Baseline coverage: Baseline activity presents us with a steady signal to detect potential disruptions in end-user connectivity, since it reduces the effect of human-triggered action. We next address

whether baseline activity is present in sufficiently many address blocks. Of the set of /24 prefixes that had any activity in any hour, within a week, or month, Figure 1b shows the CCDF of the fraction of these prefixes for which the minimum number of active IPv4 addresses in each hour is at least a given value. For example, for 44% of the /24 prefixes, the minimum number of active addresses over the course of a week is at least 40.¹ Indeed, we observe that baseline activity is not an isolated phenomenon, but that a large number of /24 address blocks show a significant minimum number of active addresses. We note that baseline activity is prevalent across many networks and addressing mechanisms (see Figure 1a for examples both of statically as well as dynamically assigned client addresses). We further study coverage of our approach in § 3.4, and see § 9.1 for discussion of IPv6.

Baseline continuity: To assess how continuous baseline activity is (and not, e.g., affected by short-term seasonal effects or frequent network restructurings), we show in Figure 1c the week-to-week change in the minimum number of active IPv4 addresses. To generate this plot, we selected all baseline values for each /24 and week in which the baseline is at least 40. We then calculate the minimum number active addresses in the subsequent week, where the latter minimum might be below 40. Figure 1c then shows the ratio of the latter minimum divided by the former baseline. Baseline activity on a per-block level is indeed very steady over time; close to 80% of the /24 address blocks show a change only in the range of $\pm 10\%$ of the active addresses, and only 2% of address blocks show changes that exceed 50% of the active addresses. Note the small peak at 0, indicating that the baseline activity changed to zero.

3.3 Detecting Activity Disruptions

Having established that active IPv4 address counts per address block remain steady over periods of time, we next introduce our approach to detect disruptions in this activity. Note that our approach focuses on offline detection of disruptions in CDN log files and we discuss the possibility of real-time analysis in § 9.1.

Figure 2 illustrates our approach for an exemplary /24 address block. For each /24 IPv4 prefix, we use a sliding window in which we calculate the minimum number of active addresses in each hour over the last 168 hours, denoted as b_0 . We advance the sliding window each hour, updating the value of b_0 . If the window reaches an hour where the number of addresses is below a threshold, $\alpha \times b_0$, for $0 < \alpha < 1$, then we tag this hour as the start of a non-steady-state period. Upon such an event, we do not advance the sliding window, and rather introduce a second, new sliding window starting at the first hour of the non-steady-state period, and calculate the minimum number of active addresses for the future 168 hours. We advance the new window until it reaches a new baseline that is at least $\beta \times b_0$, meant to be “reasonably” close to b_0 . The hour at which this occurs is the end of the non-steady-state period, and the start of a new steady-state period. We then identify a *disruption event* as those contiguous hours in the non-steady-state period where number of active addresses is lower than $b_0 \times \min(\alpha, \beta)$, shown in red in Figure 2. Typically, there is just one disruption event, though sometimes, as in Figure 2, there is more than one.

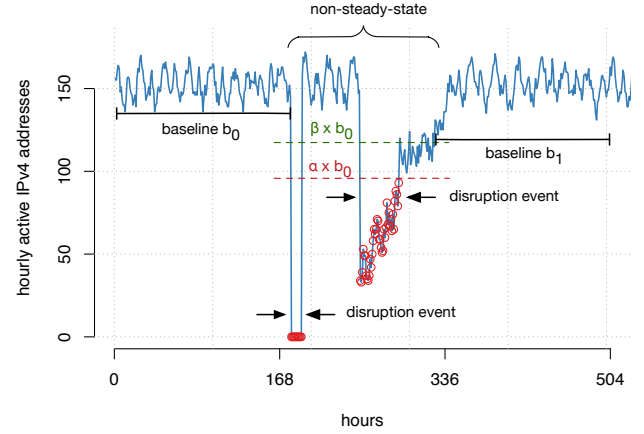


Figure 2: Disruption detection: If the number of hourly active addresses declines below a threshold $\alpha \times b_0$, where b_0 is the minimum number of active addresses in the last 168 hours, we enter a non-steady-state period. Once the minimum number of active addresses over 168 hours is restored to at least $\beta \times b_0$, the non-steady-state period ends. Within non-steady-state periods, *disruption events* are contiguous hours with fewer than $b_0 \times \min(\alpha, \beta)$ active addresses.

However, there are time series where the criterion for the new baseline is never met, or not met for a long time, possibly due to network restructuring or some long-term change. In this work, we are not interested in such events, and thus we impose a limit of two weeks for the duration of detected disruptions: If the second window advances for two weeks without satisfying the criteria for a new baseline, then we do not identify disruption events for this non-steady-state period, but continue to advance the window until the criteria for the new baseline is met, if ever. Note that this excludes the detection of outages that are longer than two weeks. The above logic intentionally restricts the set of disruptions to those with *steady baseline activity both before and after the disruption event*. This simplifies, though by no means resolves, the task of inferring which disruptions are outages, see Section 5.

3.4 Trackable Address Blocks

We chose to require that the baseline activity for a /24 prefix, b_0 , be at least 40 active addresses for us to consider it to be in a *trackable state*, i.e., we will look for a disruption in the following hour. We experimented with various values and found that 40 yields a reasonable trade-off: a lower value would include more prefixes (Figure 1b) but be more vulnerable to false disruptions, which we will elaborate on in the next section. This minimum requirement for a trackable prefix prevents detection of disruptions in address blocks where the address activity regularly reaches a lower value, for example enterprise networks with little activity during weekends, or the German university prefix in Figure 1a with a baseline of 13.

Although baseline activity is often stable over long periods of time, an address block can be trackable for some weeks but not others. To assess the overall coverage of our dataset, we now consider the full observation period of one year, and count how many /24s have a baseline b_0 of at least 40 for each hour of the year. We

¹The exemplary CCDF is for a week in March 2017 and the entire month of March; other weeks and months show the same behavior.

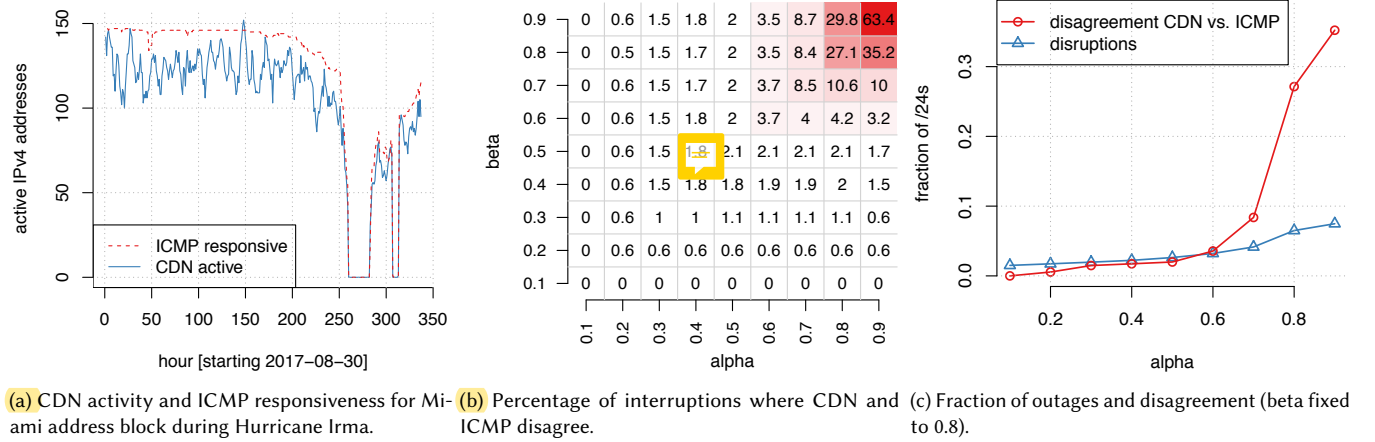


Figure 3: Tuning the robustness of our approach: Detected disruptions in CDN activity vs. ICMP-detected disruptions.

exclude the first week since we need 168 prior hours to establish the baseline. The median value across all hours is **2.3M trackable /24 address blocks in more than 12.5K ASes**. The median absolute deviation of trackable address blocks across all hours is very low: 2K /24 blocks, or 0.1%. Even during Christmas and New Year’s, the period with the lowest number of trackable blocks, the number decreases only minimally by 15K, or 0.7% of the typical 2.3M. These 2.3M trackable /24 address blocks represent 37% of all /24 prefixes that showed any activity, yet they host 82% of all active IPv4 addresses that the CDN sees and account for 80% of all requests issued to the CDN. We can, hence, track those portions of the address space that host a large majority of clients of the CDN.

3.5 Choosing Robust Parameters

We can adjust the sensitivity of our approach by setting α and β . A high α value will result in more detected disruptions—a high value will set the threshold close to the baseline, thus making it more likely to be crossed—while a low α may miss partial disruptions. The β value directly affects how sensitive the algorithm is to determining the end of non-steady-state and contained disruptions—a high value will require that the prefix activity be restored to near the original baseline, while a low value may classify long-term baseline changes (e.g., permanent network restructurings) as disruptions. We have established that baseline activity (§ 3.2) is rarely affected by users’ schedules. Still, the parameters of our approach must be set so that detected disruptions are not the result of regular variability in address activity, i.e., hosts temporarily not contacting the CDN, but still active with their IP addresses.

CDN vs. ICMP as disruption signal: In order to adjust our parameters, we calibrate against an orthogonal approach to detect disruptions, active ICMP echo probing, which should be reasonably independent of CDN address activity. We will choose parameters that rarely detect disruptions that are not clearly accompanied by a drop in ICMP responsiveness. See Figure 3a, which shows the number of IPv4 addresses in a /24 prefix that contact the CDN per hour, as well as the number of ICMP responsive addresses in this block. During the disruption in address activity, we can see a disruption in ICMP responsiveness at the same time. We manually inspected

hundreds of disruption events showing this behavior and are thus confident that this example disruption in the CDN logs indicates a disruption in connectivity to an IP address block. We next apply a method based on this observation to select α and β .

Actively probing the address space has limitations that make it impossible to comprehensively evaluate our detected disruptions against ICMP responsiveness. In particular, recent measurements show that up to about 40% of the hosts contacting the CDN typically do not respond to ICMP echo requests [48]. In addition, probing every routed IP address on a continuous basis requires substantial bandwidth for the probes, operator attention, and a strategy for reacting to firewall-based filtering of probe traffic. However, while these limitations prevent a comprehensive evaluation, we can compare *some* address blocks and time periods, for which we have available data, for the purpose of adjusting our parameters.

ICMP survey data: We leverage address space survey datasets provided by ISI [4–7] (the ICMP data shown in Figure 3a is also from [7]). ISI address space surveys periodically, every 11 minutes, send ICMP echo requests to all IP addresses within $\approx 1\%$ of the allocated IPv4 /24 address blocks. Surveyed address blocks are selected using different policies, i.e., the survey population both contains randomly selected address blocks as well as some address blocks that were responsive to ICMP requests in earlier probing attempts (see [28] for details). Hence, while this dataset covers only a small portion of the space, it comprehensively probes every address *within* that subset of /24s. We leverage data from four surveys executed between June and September 2017. In total, this dataset contains some 52K /24 address blocks, 21K probed over a two week window, and 31K over a four week window. In a first step, we remove ISI blocks that never had more than 40 responsive IP addresses in any hour, reducing our set by some 53% down to 25K blocks. Next, we intersect the 25K blocks with those address blocks that were in a trackable state in our CDN data (recall § 3.4), leaving us with 15K address blocks for comparison.

Comparing CDN and ICMP disruptions: Next, we execute our disruption detection for each combination of α and β values ranging from 0.1 to 0.9. Whenever our approach detects a disruption, we compare the time interval of our disruption with ICMP following a

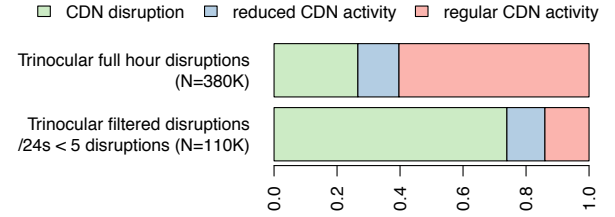
two-step approach: For those hours that were not affected by the disruption, we require that ICMP responsiveness never drops below 40 and has a maximum range of ± 30 addresses.² This ensures that we only compare address blocks for which we have a steady signal of ICMP responsiveness in its regular state. If this criterion is satisfied, we then classify the disruption into *agree* or *disagree*. We say that a disruption agrees, if the *maximum* number of ICMP responsive addresses during the disruption is smaller than the *minimum* number of ICMP responsive addresses outside the disruption. That is, at all points in time, we see more ICMP responsive addresses outside of the disruption compared to the disrupted hours themselves. Note that the number of disruptions, and of address blocks, that we compare varies depending on the individual α and β , but ranges between 200 and 2000 address blocks. We are aware that this is a comparably small sample. For this reason we strive for minimal disagreement and set strong criteria for our cross-evaluation.

3.6 Data-driven Parameter Selection

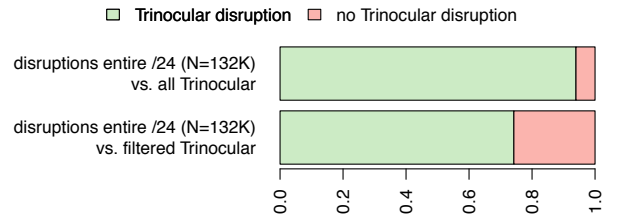
Figure 3b shows the percentage of *disagreement* between our CDN detection and ICMP for different values of α and β . For the percentages in Figure 3b, recall that the number of samples varied, and thus there is some coarseness when comparing the percentages, particularly for fractions of a percentage. Nevertheless, Figure 3b yields some general observations. Very low values of α and β exclusively capture disruptions where the number of active addresses goes to zero. For these cases, we did not detect a single instance of disagreement. With higher values, our detection sensitivity increases — up to the extreme case where both α and β are at 0.9, resulting in more than 60% of cases where ICMP responsiveness does not drop with CDN activity. To keep the disagreement below roughly 3%, α and β can not both be greater than 0.5. Also, ignoring for the moment the impact of the choice of α , a high value of β enforces a higher recovery of address activity, which leads to a more conservative, restrictive criterion for determining the termination of a disruption (i.e., lessens the likelihood that a level-shift change is falsely detected as a disruption, at the risk of missing some true disruptions). We chose β to be 0.8. Then, for $\beta = 0.8$, Figure 3c shows how the fraction of disagreement (potential false positives) as well as the fraction of address blocks in which we detect a disruption (completeness) changes for different values of α . While the number of disruptions increases only linearly up to alpha values of 0.5, the number of disagreements steeply increases for α values of 0.6 or larger. Based on our observations, we fix α to 0.5 and β to 0.8 for the remainder of this work.

With these parameters, there remain a few cases where ICMP responsiveness and CDN activity disagree, all of which were partial disruptions to address activity: not all addresses were affected. We opt for conservativeness: fewer disruptions but more confidence that they are really disruptions. While we detect all disruptions that affect an entire /24 (assuming the /24 was in a trackable state before the disruption), we will not detect all disruptions that affect parts of /24s. In the following, we note where we separate disruptions that affect entire /24s versus disruptions that only affect partial /24s. In addition to our cross-validation against ICMP responsiveness, in

²We exclude two hours directly before and after the disruption event from this comparison to account for our hourly time-binning.



(a) Trinocular-detected disruptions in the CDN logs: For 60% of detected Trinocular disruptions, address activity as seen from the CDN remains unchanged. The CDN confirms only 27% of Trinocular disruptions. Filtering out address blocks with frequent Trinocular disruptions reduces the number of Trinocular disruptions, but increases agreement significantly.



(b) CDN-detected disruptions in Trinocular: Trinocular confirms 94% of CDN-detected disruptions that affect all addresses within a /24. Filtering Trinocular data by removing frequently disrupted blocks reduces agreement and thus likely misses true disruptions.

Figure 4: Detected disruptions in the CDN logs and Trinocular, a state-of-the-art active outage detection system.

Section 5 we leverage an external dataset revealing device activity. This latter dataset contradicts our detected disruptions in less than $< 0.01\%$ of the cases, making us confident that detected disruptions indicate loss of connectivity of the concerned address blocks.

3.7 Evaluation against State-of-the-Art

Next, we evaluate our disruption detection approach against a state-of-the-art system for Internet-wide detection of outages via active probing: Trinocular [46]. We rely on a three-month dataset (2017-04-03 to 2017-07-02) made available by ISI [8]. For each /24 address block, we extract all disruptions detected by Trinocular, i.e., a *down* event for an address block followed by an *up* event. We then compare time periods of Trinocular-disrupted address blocks with disruptions detected in our CDN logs and vice versa. For both datasets, we only compare disruptions that affect address blocks that were in a *trackable* state in the other dataset at the time of the disruption (i.e., we saw a baseline greater than 40 in the CDN logs, and, likewise, a block was in an *up* state in Trinocular prior to a disruption). We say that disruptions in the two datasets *agree* if we find an, at least partial, overlapping in time of disruptions in the two datasets. In future work, we plan to conduct a more detailed analysis of timing aspects. Figure 4 shows our results.

Overall coverage: The Trinocular dataset contains information for some 3.5M /24 address blocks (after removing blocks that were in an unmeasurable state during our time window). On the first day of the comparison period, the CDN recorded activity from some

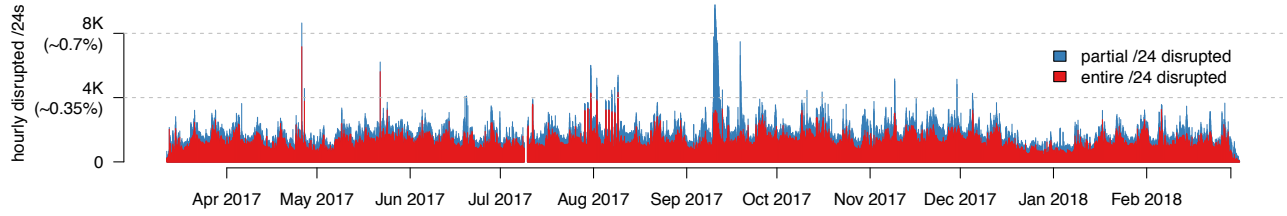



Figure 5: Hourly disrupted /24s detected over the course of our one-year observation period. Stacked bars show disruptions that affected all addresses within a /24 (red) as well as disruptions that affected only some addresses within a /24 (blue).

5.1M /24 address blocks, 2.3M of them were in a CDN-trackable state. Some 1.6M /24 address blocks are covered in both datasets.

Trinocular disruptions in CDN logs: For evaluating the visibility of Trinocular-detected disruptions in the CDN logs, we restrict the analysis to disruptions in the Trinocular dataset that span at least one calendar hour, since we can not detect shorter disruptions in the CDN logs due to binning. Some 29.9% of the disruptions in the overall Trinocular dataset span at least one calendar hour. We find that Trinocular detects significantly more disruptions compared to our CDN-detected disruptions. Figure 4a shows how Trinocular disruptions are reflected in CDN activity. We classify them into *CDN disruption*: The CDN logs show a full or partial disruption that agrees with Trinocular’s, *reduced CDN activity*: we see a decrease in the baseline in the CDN logs, but not enough to meet our criterion for a disruption, *regular CDN activity*: no decrease in the baseline, and the CDN continues to serve content. Our approach confirms only some 27% of Trinocular outages. In 60% of the cases, the baseline did not change at all during the detected disruption by Trinocular, implying a high percentage of false positive detections. **Filtering Trinocular:** We discussed this result with the authors of Trinocular, who suggested that the cause could be a known issue with their methodology, whereby Trinocular detects frequent change of state of some address blocks. We then chose a simple, first-order filter of the Trinocular dataset and only considered address blocks with fewer than 5 disruptions over the 3 month time period. This reduces the number of disruptions for comparison by more than two thirds, down to 110K, but only reduces the overall number of Trinocular-trackable blocks by some 3% (from 3.5M /24s down to 3.4M /24s). Comparing this subset against our logs, we now confirm some 74% of the detected Trinocular disruptions, though for some 26% the CDN was still serving content to at least a portion of the address block.

CDN disruptions in Trinocular: Comparing in the opposite direction, i.e. when studying the visibility of CDN-detected disruptions in Trinocular, we restrict ourselves to CDN-detected disruptions that affected all addresses in a /24 address block, since Trinocular’s design focuses on block-level disruptions and outages. Figure 4b shows that Trinocular indeed detected a disruption in some 94% of all CDN-detected disruptions. Comparing the CDN disruptions against the filtered Trinocular dataset reduces the agreement down to 74%. Thus, although filtering out Trinocular blocks with 5 or more disruptions had the benefit of significantly increasing the fraction of Trinocular disruptions that were also seen by the CDN, it has the disadvantage that the fraction of CDN-detected disruptions not seen by Trinocular increased from 6% to 26%. 

4 A GLOBAL VIEW OF DISRUPTIONS

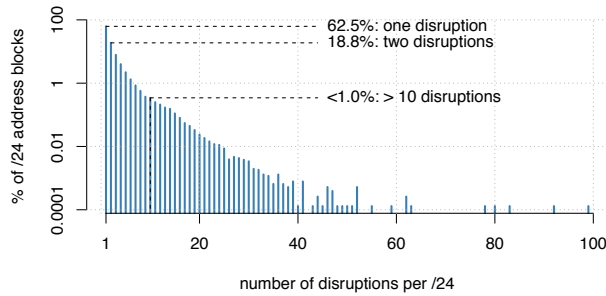
We next apply our disruption detection mechanism over the entire period of our dataset and study disruptions on a broad scale. Figure 5 shows the absolute number of disrupted /24 address blocks in each hour between March 2017 and March 2018. Here, we partition disruptions in two categories: the red bars show disruptions that affected the entire /24 (i.e., the number of active addresses during the disruption went to 0), while the blue bars (stacked) show disruptions that affected only parts of a /24 (i.e., some addresses remained active during the disruption). We can make several observations from this figure: (i) the number of disrupted /24 address blocks ranges at around 2000, or some 0.2% of tracked address blocks, with only a few major events deviating from this pattern: In September 2017, we can see a strong spike in the number of disrupted /24s (Hurricane Irma), and notice that during this event the majority of affected /24 address blocks only showed partial disruptions in address activity. Aside from several other spikes indicative of single large-scale events (§ 4.1), we observe that the number of disrupted /24 blocks follows a weekly pattern throughout the year, but that this pattern is mostly absent during the Christmas/New-Year’s period. We further investigate this phenomenon in § 4.2.

4.1 Disruption Patterns in Space

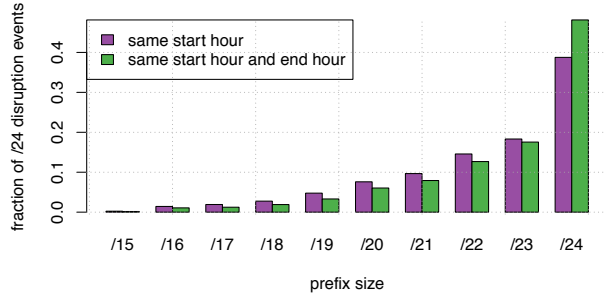
In this section, we are interested in understanding how often individual address blocks are affected by disruptions, as well as if disruptions typically span isolated address blocks or also affect neighboring prefixes at the same time.

Disruptions per /24: Figure 6a shows the distribution of disruption events per individual /24 address block. Note that we only show address blocks that had at least one disruption event during our observation period. Here, we can see that more than 60% of /24 prefixes had only a single disruption event during the entire observation period of one year. Less than 1% of /24 address blocks had 10 or more disruption events, with only a handful of prefixes having more than 20, and only 8 prefixes having more than 60 disruptions, and these 8 prefixes contain only about 0.05% of all disruption events. The important takeaway here is that the periodic behavior in Figure 5 is not the result of some recurring pattern affecting the same set of /24 address blocks. Instead, the weekly pattern affects disparate /24 address blocks.

Disruption prefix size: We next group /24-disruption events together. In a first step, we put all disruptions into time bins using two different rules: In the more relaxed case, /24 disruption events with the same start hour are placed in a bin. In the more strict case, we group /24 disruptions events together according to their start-



(a) Disruptions per /24 address block, if ever disrupted.



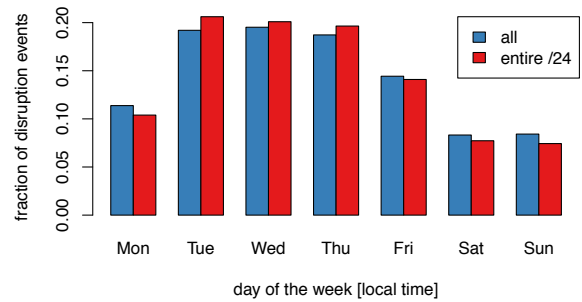
(b) Grouping detected /24 disruption events together: The majority of disruption events spans multiple adjacent /24 prefixes. In some instances, every /24 address block within an entire /15 shows a disruption.

Figure 6: Spatial properties of disruptions.

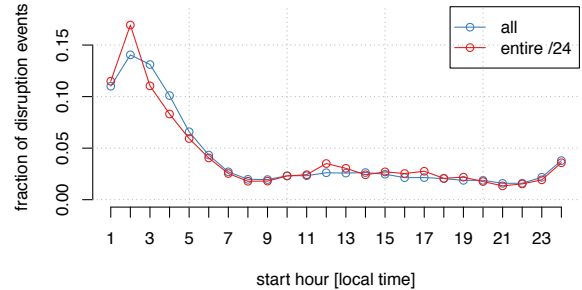
and end hour (i.e., only disruption with the same duration and start hour will be in the same bin). Then, for all /24 blocks within each bin, we group /24's that are adjacent in address space, and find the longest prefix that is completely filled by these /24s. For example, if we have four /24 prefixes that are adjacent in address space, and are contained in a /22 prefix, and the neighboring /24 prefixes would not completely fill a /21 prefix, then for these four /24 prefixes the covering prefix is a /22.

Figure 6b shows the histogram of disruption events partitioned by the largest prefix that covers individual /24 prefixes. For example, 18% of the disruption events with the same start time occur in /24 prefixes that have a /23 covering prefix, while 39% do not aggregate into a shorter prefix. We observe that with the restriction of common of start times and of end times fewer disruptions group into larger prefixes (see higher green bar at /24), yet still a majority of /24 disruption events do: 52% of events with the same start and end time aggregate into shorter prefixes (61% of events only with the same start time). Note that we find instances in which all /24s contained in an entire /15 address block show a disruption starting and ending precisely at the same time. We manually investigated large /15 events and found two of them to be related to an Iranian cellular ISP, and one other related to an Egyptian ISP. For both countries, reports of willful Internet shutdowns exist [37]. We note that such abrupt events affecting large prefixes have distinct spatial properties (red spikes in April/May in Figure 5), different from, e.g., the effect of Hurricane Irma (blue spike and recovery period in September in Figure 5).

We acknowledge that this is only a first step to study spatial properties of disruptions. Alternative ways to group /24s together might involve more advanced clustering algorithms based on event



(a) Start day of disruption events (timezone-normalized).



(b) Start hour of disruption events (timezone-normalized).

Figure 7: Time patterns of disruption events.

timings [29] or alternative topological aspects, such as last visible router on traceroutes towards /24s.

4.2 Disruption Patterns in Time

Figure 5 shows an intriguing pattern: Over the course of the year, we clearly observe some recurring day-of-the-week pattern, which is less pronounced in the Christmas/New-Year's week. To better understand this pattern, we next study when disruption events typically happen. To determine the local time of disruption events, we first geolocate all our disruption events using the CDN's geolocation database. Leveraging geolocation with timezone information, we can get a good estimate of the *local* time of disruption events. Figure 7a shows a breakdown of the weekday on which we see the start of disruption events, where “entire /24” means all addresses in the prefix had no activity, and “all” also includes prefixes where some addresses still showed activity. Complementing Figure 7a, Figure 7b shows the distribution of disruption start times across hours-of-the-day.

Scheduled Maintenance: Surprisingly, we observe that disruptions are much more pronounced on weekdays, particularly Tuesday, Wednesday, and Thursday, the typical maintenance window. ~~Comarella et al. found similar results in BGP [21] and so did Beverly et al. when studying reboots of ISP routers [14].~~ The picture sharpens even more when looking at the hour of the day of these events, as shown in Figure 7b. Here, we see that most disruptions start after midnight local time, typically between 1AM and 3AM. These start times correspond with the maintenance window of major ISPs (e.g., [19, 24]). In fact, disruptions during the maintenance window dominate for many ISPs. We return to this observation in Section 8, when discussing properties of residential US ISPs. We

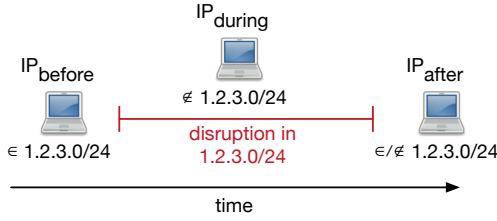


Figure 8: For detected disruptions, we check if and when a user device that was previously active in a disrupted address block is active next, during and after the disruption.

note, however, that this is not an isolated phenomenon, many ISPs across the globe show this regular disruption pattern.

ISP feedback: We shared with a contact at a major US cable provider the disruption events that we had detected in their network. They reported that all of the events that they researched corresponded to scheduled network maintenance.

5 A DEVICE VIEW OF DISRUPTIONS

Having studied macroscopic properties of disruption events in the prior section, we now shift our perspective and study disruption events from a device-centric perspective, leveraging an orthogonal dataset that allows us to track activity of individual devices across the address space before, *during*, and after disruption events. This allows us to study aspects of user mobility, as well as to identify instances in which disruptions are not indicative of service outages.

5.1 Device Activity across Address Blocks

To study activity of individual devices, we next leverage an orthogonal dataset: Logs from a service offered by the CDN to content owners whereby end users can elect to install software that will improve the performance the client experiences when accessing the content through the CDN.³

Pinpointing devices: The software runs on Windows and Mac OS X and is installed on desktops and laptops, but not smartphones. Devices with the software installed repeatedly contact the CDN, and identify themselves with the unique identifier of the software installation on the machine, herein called the “software ID,” or simply “ID”. For the present study, the relevant fields of the log lines are: the timestamp at which the log line is created, the public IP address seen by the CDN’s infrastructure at this time, and the ID. These logs are distinct from those used for the time-series-of-hits dataset of Section 3.1, and are generated only for clients that have installed the software, and their frequency varies. Thus, while a log line evidently shows that a device was active at a given timestamp with a given IP address, the absence of a log line does not imply that the device did not have Internet connectivity. We next leverage this dataset to study further attributes of identified disruptions.

Pairing devices and disruptions: We isolate only disruption events that affected entire /24 address blocks (i.e., no IP address showed any activity during the disruption) and identify all IDs that were active in the disrupted /24 address block within the last hour

³This client-installed software does not access the data of any other applications on the device.

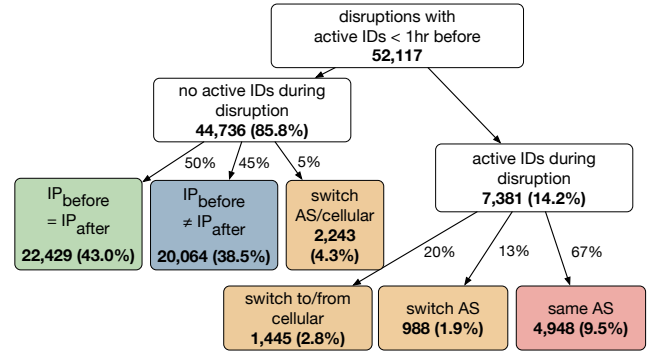


Figure 9: User devices that were previously in a disrupted address block were sometimes active in other address blocks during the disruption event. While some of these cases can be attributed to mobility or tethering, the majority had a new address in the same AS as the disrupted address block.

before the disruptions’ start time, illustrated in Figure 8. Of the 883K such disruption events, we found an ID active up to an hour before the disruption in 52K (5.9%) instances. We note that last IP address with which the device was active as IP_{before} . Next, we check if the ID is seen *during* the disruption event from some other address block, then we call the first address associated with the ID during the disruption IP_{during} . Lastly, we note the first IP address after the disruption event as IP_{after} . We next proceed and study the interplay of IDs and IP addresses during detected disruption events. Figure 9 shows our results.

Cross-validation of detected disruption events: We found only 6 instances ($< 0.01\%$) in which an ID was seen during a disruption with IP_{during} within the disrupted address block. This again shows that our disruption detection mechanism (§ 3) is effective in detecting loss of Internet connectivity for certain address blocks and does not falsely identify disruptions of address blocks that still have Internet connectivity. We omit these 6 instances from our dataset.

5.2 No Device Activity during Disruptions

We first focus on disruptions in which we did not record any intermediate activity, i.e., IP_{during} does not exist. This is the expected case, since we naturally presume that devices in disrupted address blocks lost Internet connectivity. Indeed, the majority (some 86% of our disruptions) show this behavior. While for these disruptions we do not have any indications that suggest other than a service outage (e.g., devices cannot connect to the Internet any longer), we further group these instances into whether the IP address corresponding to the software ID has changed ($IP_{\text{before}} \neq IP_{\text{after}}$) or remained the same ($IP_{\text{before}} = IP_{\text{after}}$), see Figure 9. This distinction is important, since it gives us a different level of confidence when interpreting such disruptions: If the IP address of an ID remained the same before and after the disruption, it is unlikely that the device was temporarily assigned a different address from the ISP and switched back to its original one [42]. We are, thus, more certain that such disruptions are service outages, as opposed to prefix migration events (next Section). If the address changed, however, we have lower confidence when distinguishing between device movement, address re-assignment, and service outages. We will consider

this property when assessing ways to distill service outages from disruptions in § 7.

5.3 Device Activity during Disruptions

Next, we shift our attention to the more unexpected, yet prevalent (some 14%) case: Instances of disruptions, in which activity was recorded *during* the disruption period. Note that our records reflect a lower bound of activity during disruptions, since the software does not necessarily contact the CDN during a disruption event, even if the concerned end-host has Internet connectivity. This activity can happen as a result of a few scenarios:

Mobility and tethering: First, users can physically move to a different location and connect their device to a different network, or the device is multi-homed, perhaps tethered through a cellular network. We refer to these instances as *mobility*, highlighted in orange in Figure 9. To identify such cases, we study if the switch from IP_{before} to IP_{during} involved a switch from or to a cellular address block⁴ or switched AS numbers. We note that in some 20% of the cases users switched to a cellular network, and in another 13% to a different ASN. While these cases do not reveal whether the detected disruption in the original address block resulted in a loss of service connectivity, they highlight that today a significant fraction of end users are multi-homed in the sense of having the ability to switch between different access networks, in the case of a potential network service outage.

Address reassignment: Second, the user continues to use the same Internet service provider, but the public IP address through which the user's device connects to the Internet has changed. Thus, we detect a sudden absence of all activity within the original address block, and see activity from the very same hosts from different address blocks in the same AS during the disruption. Note that this is by far the most common case for during-disruption activity, accounting for some 67% of during-disruption activity instances and for almost 10% of all detected disruption events for which we have device-specific information. While in the case of disruptions without activity (§ 5.2) as well as in cases of mobility and tethering we do not gain hard evidence on whether a disruption really resulted in a service outage or not, we can infer for these 9.5% of disruptions that they are likely not the result of a service outage. We corroborate our findings that some disruptions are not service outages in Section 6, where we identify that instances of during-dip-activity often go along with an upsurge in overall address activity in the prefix a device moves to.

6 DISRUPTIONS AND ANTI-DISRUPTIONS

Our device-centric analysis of disruptions revealed that in some 10% of disruptions, devices do not lose access service. We are next interested in the interplay between such disrupted address blocks and those address blocks into which devices move, the *alternate* /24 address blocks.

Microscopic anti-disruptions: We hence return to our activity-per-/24 timeseries and inspect both the disrupted /24 as well as the alternate /24, identified using our software ID dataset (§ 5.3). Figure 10 shows such an example. Here, we plot activity of the

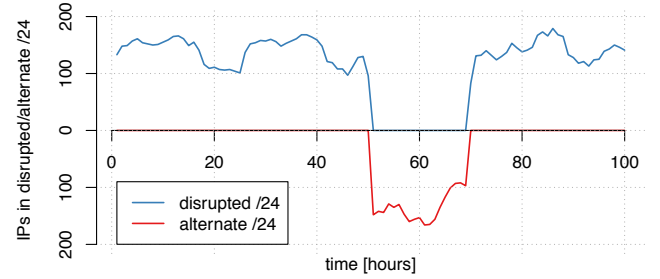


Figure 10: Example of an anti-disruption event: active IDs show activity during the disruption, but from a different address block. Their activity is correlated.

disrupted /24 in blue, and the address activity of the alternate /24 in red in the negative y -direction. Indeed, we can clearly observe patterns of alternating activity between the concerned address blocks. We refer to this phenomenon as *anti-disruption*, temporary spikes in address activity in address blocks. We note that while Figure 10 shows a clear anti-disruption signature, often the shift on an individual /24 basis is not so clear, but may become apparent when viewed network-wide.

Network-wide anti-disruptions: We next seek to leverage our observations about anti-disruptions and generalize our approach for detecting them without the need to track individual devices. To do so, we invert our disruption detection mechanism (recall § 3.3) to detect anti-disruptions. Instead of calculating the minimum number of active addresses over the prior week window, we now calculate the maximum number of active addresses. We then set our α value to 1.3 and β to 1.1.⁵ Thus, we now detect address blocks that show irregularly high activity over short periods of time. We next apply our mechanisms over the entire dataset and study disruptions and anti-disruptions on a per-AS level.

To visualize and correlate the magnitude of disruptions and anti-disruptions, for each disruption, we calculate the number of disrupted addresses in the /24 prefix as the difference between the median number of active addresses in the week prior to the disruption and the median number of addresses active during the disruption. We then assign this number to each hour that the disruption existed. Lastly, for each hour in the observation period, we sum over the number of disrupted addresses, if any, for all disruptions observed in a given AS. We do the analogous computation for anti-disruptions. Figure 11 shows three example ASes, with different levels of correlation for disruptions and anti-disruptions. While the US ISP in Figure 11a shows virtually zero correlation between disruptions, the Spanish ISP in Figure 11b shows moderate correlation, The Uruguay ISP (bottom) shows that most disruptions and anti-disruptions in this AS align very clearly. We calculate the pearson correlation across disruptions and anti-disruption (see Figure 11) to express the degree of correlation for individual ASes. **ISP feedback:** We were surprised to see the anti-disruption pattern very strongly for some ISPs, having the potential to heavily skew AS-based analysis of Internet reliability. Contacts from two ISPs, one cable and one DSL, confirmed that reassigning prefixes is a

⁴We leverage the dataset and method described in [51] to identify cellular address blocks.

⁵We experimented with various values. No combination catches all cases we observed when manually studying anti-disruption behavior on a per-AS basis.

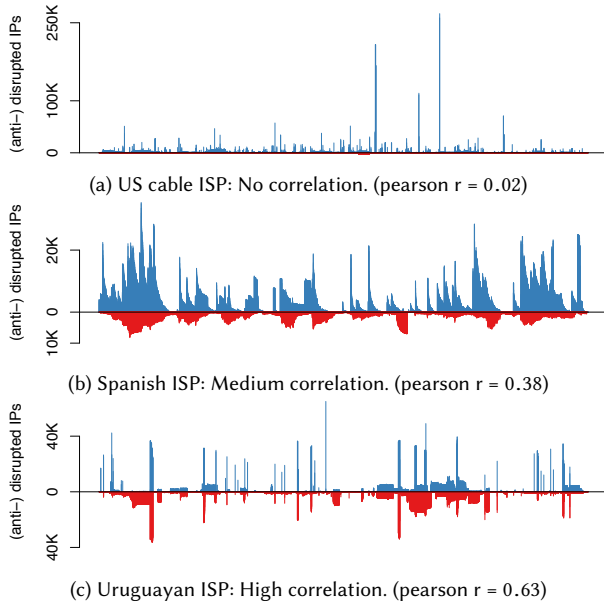


Figure 11: Hourly disrupted (blue, positive y -direction) and anti-disrupted (red, negative y -direction) IP addresses over the course of one year shows the AS-wide interplay of disruptions and anti-disruptions.

common practice. For example, to manage capacity, cable providers will move an end user base from one interface on a Cable modem termination system (CMTS) to another, triggering a renumbering of addresses. If DHCP is used to assign addresses, there is a standard procedure for doing so described in RFC 3203 [30], which defines the DHCP message *FORCERENEW*, which can be used for “Network renumbering: Under tightly controlled conditions, the *FORCERENEW* procedures can be used to brute force the renumbering of entire subnets, client per client, under control of a DHCP server.”

7 TOWARDS DISTILLING OUTAGES FROM DISRUPTIONS

Next, we study features of disruptions to determine to what extent it is possible to distinguish disruptions that reflect actual *service outages* vs. disruptions as result of prefix migration. We do so on a *per-network level*, and on the level of individual disruption events.

7.1 Network-Based Discrimination

Here we expand the per-AS classification begun in Section 6, which introduced the correlation of the time series of number of disrupted IP addresses versus anti-disrupted addresses. In addition to the correlation, we leverage information from disruptions for which we have detailed device information (recall § 5). We select 201 ASes, for which we have at least 50 disruptions with device information.

In Figure 12, for each AS, we show its pearson correlation (x -axis), as well as the fraction of disruptions (with device information) that showed interim device activity (y -axis). ASes close to the origin show both a very low correlation of disruptions and anti-disruptions and very few disruptions that had interim activity. For these ASes,

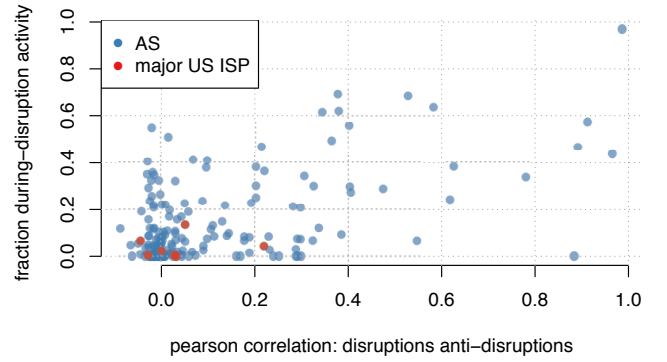


Figure 12: Per AS: Fraction of disruption with activity during the disruption vs. pearson correlation of AS-wide disruptions and anti-disruptions. Major US ISPs are highlighted and discussed in § 8.

disruptions are, by our metrics, more likely to correspond to service outages. The majority of ASes falls close to the origin: Some 54% of the ASes have both correlation lower than 0.1 and less than 10% of instances of during-disruption activity (70% have values lower than 0.2 / 0.2). However, we also find that some ASes show high anti-disruption correlations and high shares of disruptions that are not service outages (per our ID dataset). These ASes have the potential to significantly bias measurement results and, in the case of large ASes, even skew per-country assessments of Internet reliability.

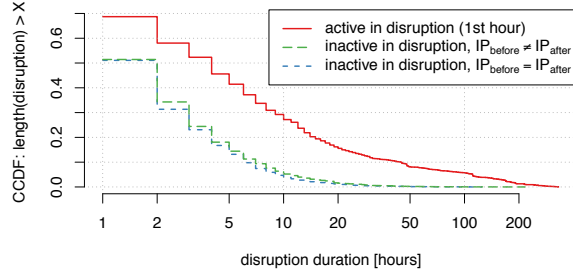
ISP feedback: When we aggregated disruption events to countries (not reported here), a smaller European country showed the worst reliability, by far, if one assumed that all disruptions were service outages. However, the cause was a major ISP in that country making extensive use of temporary reassignment of address space, resulting in major and frequent anti-disruptions. A contact at that ISP confirmed that indeed this was the practice, and that subscribers did not lose Internet access service during these events.

7.2 Feature-Based Discrimination

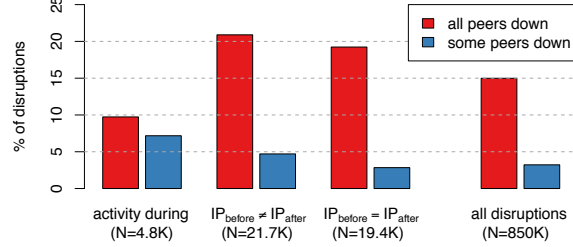
Having identified that anti-disruptions are particularly pronounced for specific networks, we next seek to assess whether there are dominant features of disruption events that allow us to distinguish between different types of disruptions. We hence study properties of disruption events for which we have per-device information (§ 5). We group disruptions into: (i) disruptions that showed activity in the same AS and are thus unlikely to represent service outages, and (ii) disruptions that did not show activity. We further partition the latter into disruptions where end-device’s IP addresses changed or not (§ 5.2). We focus on the duration of disruptions and on their visibility (or lack thereof) in the global routing table.

Disruption duration: Figure 13a shows the CCDF of the duration of disruptions, where we distinguish between our three classes. Disruptions for which we recorded interim device activity⁶ last, on average, longer than disruptions for which there was no device activity. This effect becomes particularly pronounced for disruptions that last longer than ≈ 20 hours, where the fraction of disruptions

⁶Here, we restrict our set of disruptions to only consider those in which activity was recorded in the first hour to avoid bias towards longer disruptions.



(a) Duration of disruption events. Disruptions that do not cause end users to lose connectivity are more likely to last longer than disruptions that result in an actual service outage.



(b) BGP visibility of disruptions and outages. Some 18% of disruptions are reflected w/ BGP withdrawals. A BGP withdrawal does not imply an outage: In about 16% of the disruptions that had activity during the disruption, we still see a BGP withdrawal.

Figure 13: Properties of different types of disruptions.

without activity becomes very small. This observation can prove helpful for outage detection system (ours included) when determining the maximum possible duration of detection intervals. We note, however, that also some 30% of disruptions with interim activity last just one hour. Looking at the two cases of disruptions without interim address activity, we note that there is little to no difference between instances where the device’s IP addresses changed vs remained unchanged after the disruption.

Visibility in BGP We next assess if our detected disruptions are reflected in global routing table activity, i.e., whether they align with BGP withdrawals. While earlier work has shown that BGP withdrawals do not necessarily imply loss of connectivity due to the existence of default routes [18], we assess to what extent edge activity disruptions and outages are reflected with BGP withdrawals. We selected 10 large and geographically diverse ASes that provide a full BGP feed to RouteViews.⁷ We then process weekly Routeviews dumps and the subsequent updates and tag each /24 and hour during our time period with the following BGP state: Number of peers that saw a route to the corresponding /24 address block, and number of peers that did not see a route to the corresponding /24 address block (using longest prefix matching). Note that both numbers can be at max 10, and it is possible for a prefix to be both visible and invisible in a certain hour.

Then, for each disruption that resulted in a complete loss of activity, we get the BGP state before the disruption (2 hours before the first disrupted hour) as well as during the disruption (first hour of the disruption). We only consider disruptions for which at least

⁷ASes: 3356,1221,13030,2497,286,2914,6539,6762,6939,7018.

	U.S. Cable ISPs			U.S. DSL ISPs			
	ISP A	ISP B	ISP C	ISP D	ISP E	ISP F	ISP G
anti-disruption corr.	0.22	0.029	-0.027	0.033	0.002	-0.043	0.052
disrupt. w/activity	3.9%	0.5%	0.5%	0.0%	2.6%	6.5%	14.3%
ever disrupted /24s	22.4%	45.1%	36.8%	8.0%	30.2%	12.4%	25.3%
only hurricane*	11.3%	0.9%	2.3%	22.5%	1.3%	0.2%	2.9%
only maintenance†	67.3%	54.0%	74.9%	28.4%	59.6%	71.2%	62.2%
median disruptions*	1	1	1	1	1	1	1

★: /24s disrupted only in week 2017-09-09 to 2017-09-15.

†: /24s disrupted only weekdays 12AM - 6AM, excluding hurricane period.

*: median disruptions per /24 only for /24 with at least one disruption.

Table 1: US broadband ISPs typically show few indications of anti-disruptions. The majority if their address space saw either zero or a single disruption, the majority of disrupted /24s were exclusively affected during maintenance hours.

9 peers saw the prefix before the disruption (we removed some 3% of disruptions in which this was not the case). We then tag a disruption as *all peers down*, if at some point during the first hour of the disruption all peers lost visibility to the prefix. We tag it as *some peers down*, if the number of peers that did see the announcement was lower than before the disruption, but not zero.

Figure 13b shows how our detected disruptions (in the different classes of disruptions) correspond with BGP withdrawals. Note that only about 25% of the disruptions that had no sign of activity during the disrupted period (i.e., are more likely to be a real service outage) coincided with a BGP withdrawal (either all-peers-down or some-peers-down). Thus, about 75% of these disruptions were not evident from BGP. Whether the address changed after the disruption only has a minimal (but visible) effect. Moreover, the left two bars of Figure 13b show that some 16% of the disruptions that had interim device activity, indicating that the disruption was not a service outage, still coincided with BGP withdrawals. Interestingly, a higher proportion of these withdrawals were not visible to all BGP peers. Thus, when leveraging BGP withdrawals as outage detection signal: *withdrawal and absence of a prefix from the global routing table is not definitive of a service outage.*

8 CASE STUDY: U.S. BROADBAND

We next illustrate our findings with a case study of major US ISPs. We selected the 7 largest US broadband ISPs, covering the majority of US broadband subscribers [17]. Table 1 shows our results. All of the ISPs are well-represented in our dataset.

Disruptions vs. outages: For each of these ISPs, Table 1, top lines, reports their anti-disruption behavior, as well as the percent of disruptions for which activity was observed during the disruption, § 5. We also annotated these ISPs in Figure 12. We note that with the exception of ISP A and ISP G, most major US ISPs do not show strong indicators of disruptions as result of mass prefix migrations (§ 6). ISP A shows a higher correlation of anti-disruptions and disruptions, while ISP G shows a higher percentage of disruptions in which we detected activity in other address blocks (§ 5.3). While disruptions cannot be taken “at face value” to be service outages, these ISPs are not among those that can heavily skew results. (e.g., top-right region in Figure 12).

Total disruptions: For the active /24s during the one-year observation interval, we see a very heterogeneous picture for the percent

that saw a disruption event, ranging from some 8% up to some 45%. We caution against interpreting this number in favor of individual ISPs, since the number of disrupted /24s depends on a variety of factors dependent on individual network management practices, such as filling degree (subscribers per /24), as well as churn in address block use [48]. We note that in all cases less than half of their active address space saw a disruption.

Effect of Hurricane Irma: Zooming in on those /24s that ever had a disruption, we find that Hurricane Irma [36] was the most profound event for ISPs active in the Florida region. Some 22.5% (ISP D) resp. 11.3% (ISP A) of all disrupted /24s were only disrupted within this very week, out of a total time period of 54 weeks.

Scheduled maintenance: Strikingly, we find that for all but one of the ISPs, the majority of ever disrupted /24s was exclusively disrupted during the typical maintenance window, weekdays between midnight and 6AM. For three of our ISPs, some 70% of all disrupted /24s fall only within this time period.

9 DISCUSSION AND OUTLOOK

Our study reveals a set of observations that challenge common practice and knowledge, and yield implications and future directions for outage detection and interpretation.

9.1 Implications for Outage Detection

We challenge a core assumption that disruptions—temporary loss of connectivity of individual address blocks—are necessarily indicative of service outages. We found evidence of bulk reassignment of IP addresses, sometimes resulting in large-scale anti-disruptions events; these have the potential to confuse outage detection mechanisms, including ours, as well as prior work [22, 46, 54]. Leveraging our device-specific dataset, we find that some 10% of disruptions are the result of such migrations. Since anti-disruption behavior is highly unevenly distributed across different ASes in different regions of the world, this phenomenon can easily lead to severe over-estimations of Internet outages when attempting to study reliability in individual regions, or networks. Moreover, we find that state-of-the-art active outage detection overestimates disruption occurrences, resulting from a few unstable blocks. After filtering, we confirm the majority of Trinocular-detected disruptions, boding well for further research and refinement of active outage detection. Further, we find that outages at the edge are hardly visible in the global routing table, with only some 20%-25% of disruptions that are very likely outages resulting in a loss of BGP visibility. Contrarily, we find that even a BGP withdrawal is not a definitive indication of a service outage, either. Some 15% of disruptions that do not result in service outages show up with BGP withdrawals. Our findings caution against taking such measurement results at face value.

Future directions: With the proliferation of Smartphone use, as well as smart home devices, baseline activity is likely to increase in the future, further expanding the coverage for passive outage detection. Other vantage points (e.g., traffic at border routers of ISPs or universities) could potentially capture such activity at a finer granularity in space and time, albeit with a smaller coverage. More fine-grained measurements could allow for better matching of disruptions and anti-disruptions, potentially allowing to isolate and remove such cases from outage detection analyses. It remains

an open question whether it is feasible to detect such instances with active techniques, since it would require probing vast ranges of often inactive parts of the address space.

It is currently unclear how increasing deployment of Carrier-Grade NAT gateways as result of IPv4 exhaustion [47, 49] might affect address-based outage detection systems, including ours. In the IPv6 Internet, passive approaches to track edge reliability will become more important, where active probing is problematic due to the vastness of the space and the ephemeral nature of addresses [44]. We plan to evaluate the feasibility of our approach for IPv6 traffic. A key challenge here will be to identify address aggregates, prefixes, that yield a baseline activity, where the size of these prefixes will necessarily vary greatly across the client address space, see [45]. An essential feature we leverage for disruption detection is constant baseline activity before and after a disruption event, which does not allow for online analysis. While we can certainly estimate the start of a potential disruption, online analysis can not immediately distinguish between temporary events (disruptions) vs. long-term changes, and level shifts. We plan to investigate such events, their prevalence, and impact on online analysis. To detect disruptions in prefixes where activity regularly goes to near zero, say on weekends, the notion of baseline could be generalized to a not necessarily contiguous set of measurement bins.

9.2 Implications for Outage Interpretation

We have learned that disruptions in address activity can have a variety of causes, and that planned human intervention is a major factor. We are able to identify likely causes for many of the detected disruptions, including service outages. For example, consider ISP A in our case study (§ 8): some 67% of /24s were only disrupted during scheduled maintenance intervals, another 11% only during a one-week interval of Hurricane Irma. That leaves only some 20% of disrupted blocks that fall outside these two categories. A key implication here is that the interpretation of reliability and outage measurements must take such factors into account and be qualified by specific questions under study. Does a service outage during scheduled maintenance have the same significance as one due to an unplanned network fault? Should Service Level Agreements (SLAs) make a distinction? SLAs for enterprise Internet connectivity (e.g., [1, 2]), for example, exclude service outages within scheduled maintenance intervals from network availability calculations (albeit sometimes with a clause that subscribers be notified of such events), as well as service outages caused by events of *force majeure* (e.g., natural disasters). Thus, statistics on disruptions and outages need to be put into proper perspective.

Future directions: Accurate measurement and interpretation of Internet outages will become more critical in the future, when ISPs will most likely become subject to more stringent regulations e.g., by the FCC in the US, in particular outage reporting requirements. Current reporting requirements [3] cover telephone service and set reporting criteria based on minimum duration of outages (30 minutes) and affected user minutes (900,000 user minutes). A key challenge will be how to define criteria for Internet outages, their duration, magnitude, and eventual impact on end users to derive robust threshold criteria for Internet outage reporting requirements.

ACKNOWLEDGMENTS

We thank our shepherd John Heidemann and the anonymous reviewers for their thoughtful feedback and the Custom Analytics group in Akamai for their support. This work was partially supported by the MIT Internet Policy Research Initiative, William and Flora Hewlett Foundation grant 2014-1601, and NSF grant CNS-1619048.

REFERENCES

- [1] AT&T Switched Ethernet Service Guide. Section 3 - Service Level Agreement. <http://cpr.att.com/pdf/se/0001-0003.pdf>.
- [2] Comcast Business: Enterprise Dedicated Internet PSA. https://business.comcast.com/terms-conditions-ent/enterprise_dedicated-internet-psa.
- [3] FCC. 47 CFR Part 4 -DISRUPTIONS TO COMMUNICATIONS. Outage reporting requirements - threshold criteria. <https://www.law.cornell.edu/cfr/text/47/part-4>.
- [4] Internet Addresses Survey dataset, PREDICT ID: USC-LANDER/internet-address-survey-reprobing-it76c-20170723/rev7956. Traces taken 2017-07-23 to 2017-08-06. Provided by the USC/LANDER project. <http://www.isi.edu/ant/lander>.
- [5] Internet Addresses Survey dataset, PREDICT ID: USC-LANDER/internet-address-survey-reprobing-it76w-20170628/rev7942. Traces taken 2017-06-28 to 2017-07-13. Provided by the USC/LANDER project. <http://www.isi.edu/ant/lander>.
- [6] Internet Addresses Survey dataset, PREDICT ID: USC-LANDER/internet-address-survey-reprobing-it77c-20170914/rev8018. Traces taken 2017-09-14 to 2017-09-29. Provided by the USC/LANDER project. <http://www.isi.edu/ant/lander>.
- [7] Internet Addresses Survey dataset, PREDICT ID: USC-LANDER/internet-address-survey-reprobing-it77w-20170830/rev8013. Traces taken 2017-08-30 to 2017-09-14. Provided by the USC/LANDER project. <http://www.isi.edu/ant/lander>.
- [8] Internet Outage Dataset, PREDICT ID: USC-LANDER/internet-outage-adaptive-a28all-20170403. Provided by the USC/LANDER project. <http://www.isi.edu/ant/lander>.
- [9] Charu C. Aggarwal. *Outlier Analysis, second edition*. Springer Publishing Company, Incorporated, 2016.
- [10] O. Argon, A. Bremner-Barr, O. Mokryn, D. Schirman, Y. Shavitt, and U. Weinsberg. On the dynamics of IP address allocation and availability of end-hosts. *arXiv preprint arXiv:1011.2324*, 2010.
- [11] R. Banerjee, A. Razaghpanah, L. Chiang, A. Mishra, V. Sekar, Y. Choi, and P. Gill. Internet Outages, the Eyewitness Accounts: Analysis of the Outages Mailing List. In *PAM*, 2015.
- [12] K. Benson, A. Dainotti, K. Claffy, A. Snoeren, and M. Kallitsis. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *ACM IMC*, 2015.
- [13] R. Beverly and M. Luckie. The Impact of Router Outages on the AS-level Internet. In *ACM SIGCOMM*, Aug 2017.
- [14] R. Beverly, M. Luckie, L. Mosley, and K. Claffy. Measuring and Characterizing IPv6 Router Availability. In *Passive and Active Network Measurement Workshop (PAM)*, pages 123–135, Mar 2015.
- [15] Z. Bischof, F. Bustamante, and N. Feamster. The Growing Importance of Being Always On – A First Look at the Reliability of Broadband Internet Access. In *Research Conference on Communications, Information and Internet Policy (TPRC)* 46, 2018.
- [16] Z. Bischof, F. Bustamante, and R. Stanojevic. Need, Want, Can Afford: Broadband Markets and the Behavior of Users. In *ACM IMC*, 2014.
- [17] BroadbandNow. The Complete List of Internet Providers in the US. <https://broadbandnow.com/All-Providers>.
- [18] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *ACM IMC*, 2009.
- [19] Comcast Business. Maintenance Notifications. <https://business.comcast.com/terms-conditions-ent/maintenance>.
- [20] R. Cleveland, W. Cleveland, and I. Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3, 1990.
- [21] G. Comarella, G. Gürsun, and M. Crovella. Studying interdomain routing over long timescales. In *ACM IMC*, 2013.
- [22] A. Dainotti, C. Squarcella, E. Aben, K. Claffy, M. Chiesa, M. Russo, and A. Pescape. Analysis of Country-wide Internet Outages Caused by Censorship. In *ACM IMC*, 2011.
- [23] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. NetDiagnoser: Troubleshooting Network Unreachabilities Using End-to-end Probes and Routing Data. In *CoNEXT*, 2007.
- [24] DSLReports.com. Is there an official DSL network maintenance window? <http://www.dslreports.com/faq/2496>.
- [25] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *USENIX Security Symposium*, 2013.
- [26] V. Giotas, C. Dietzel, G. Smaragdakis, A. Feldmann, and E. Aben. Detecting Peering Infrastructure Outages in the Wild. In *ACM SIGCOMM*, 2017.
- [27] S. Grover, M. Park, S. Sundaresan, S. Burnett, H. Kim, B. Ravi, and N. Feamster. Peeking behind the NAT: an empirical study of home networks. In *ACM IMC*, 2013.
- [28] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Banister. Census and survey of the visible internet. In *ACM IMC*, 2008.
- [29] J. Heidemann, Y. Pradkin, and A. Nisar. Back out: End-to-end inference of common points-of-failure in the internet (extended). Technical Report ISI-TR-724, USC/Information Sciences Institute, Feb 2018.
- [30] C. Hublet and R. De Schrijver. DHCP reconfigure extension. IETF RFC 3203.
- [31] V. Jandhyala, S. Fotopoulos, I. MacNeill, and P. Liu. Inference for single and multiple change-points in time series. *Journal of Time Series Analysis*, 34(4):423–446, 2013.
- [32] U. Javed, I. Cunha, D. R. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy. PoiRoot: Investigating the Root Cause of Interdomain Path Changes. In *ACM SIGCOMM*, 2013.
- [33] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying Black Holes in the Internet with Hubble. In *NSDI*, 2008.
- [34] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical Repair of Persistent Route Failures. In *ACM SIGCOMM*, 2012.
- [35] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *ACM SIGCOMM*, 2000.
- [36] Miami Herald. No internet after Irma means no work and no fun. When will I be online again? <http://www.miamiherald.com/news/weather/hurricane/article173954151.html>.
- [37] Al Jazeera News. Rising Internet shutdowns aimed at 'Silencing Dissent'. <https://tinyurl.com/y8pb6eq9>.
- [38] Broadband in the U.K.: data and research. <https://www.ofcom.org.uk/research-and-data/telecoms-research/broadband-research>.
- [39] Broadband Measurement Project, Canada. <https://crtc.gc.ca/eng/internet/proj.htm>.
- [40] Measuring Broadband America. <https://www.fcc.gov/general/measuring-broadband-america>.
- [41] Measuring Broadband Australia. <https://www.accc.gov.au/consumers/internet-phone/monitoring-broadband-performance>.
- [42] R. Padmanabhan, A. Dhamdhere, E. Aben, K. Claffy, and N. Spring. Reasons Dynamic Addresses Change. In *ACM IMC*, 2016.
- [43] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, 1997.
- [44] D. Plonka and A. Berger. Temporal and Spatial Classification of Active IPv6 Addresses. In *ACM IMC*, 2015.
- [45] D. Plonka and A. Berger. kIP: a Measured Approach to IPv6 Address Anonymization. *CoRR*, abs/1707.03900, 2017.
- [46] L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding Internet Reliability Through Adaptive Probing. In *ACM SIGCOMM*, 2013.
- [47] P. Richter, M. Allman, R. Bush, and V. Paxson. A Primer on IPv4 Scarcity. *ACM CCR*, 45(2), Apr 2015.
- [48] P. Richter, G. Smaragdakis, D. Plonka, and A. Berger. Beyond Counting: New Perspectives on the Active IPv4 Address Space. In *ACM IMC*, 2016.
- [49] P. Richter, F. Wohlfart, N. Vallina-Rodriguez, M. Allman, R. Bush, A. Feldmann, C. Kreibich, N. Weaver, and V. Paxson. A Multi-perspective Analysis of Carrier-Grade NAT Deployment. In *ACM IMC*, 2016.
- [50] RIPE NCC. Atlas. <http://atlas.ripe.net>.
- [51] John P. Rula, Fabián E. Bustamante, and Moritz Steiner. Cell Spotting: Studying the Role of Cellular Networks in the Internet. In *ACM IMC*, 2017.
- [52] SamKnows. Test methodology white paper, 2011.
- [53] M. A. Sánchez, 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.
- [54] A. Schulman and N. Spring. Pingin' in the Rain. In *ACM IMC*, 2011.
- [55] A. Shah, R. Fontugne, E. Aben, C. Pelsner, and R. Bush. Disco: Fast, good, and cheap outage detection. In *TMA*, 2017.
- [56] Y. Shavitt and E. Shir. DIMES: Let the Internet Measure Itself. *SIGCOMM Comput. Commun. Rev.*, 35, October 2005.
- [57] D. A. Stephens. Bayesian retrospective multiple-changepoint identification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 43(1):159–178, 1994.
- [58] S. Sundaresan, S. Burnett, N. Feamster, and W. Donato. BISmark: A testbed for deploying measurements and applications in broadband access networks. In *USENIX ATC*, 2014.
- [59] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *ACM SIGCOMM*, 2010.
- [60] O. Vallis, J. Hochenbaum, and A. Kejariwal. A Novel Technique for Long-Term Anomaly Detection in the Cloud. In *Usenix HoutCloud*, 2014.