ABSTRACT

Title of dissertation: ANALYZING INTERNET RELIABILITY REMOTELY WITH PROBING-BASED TECHNIQUES

Ramakrishna Padmanabhan
Doctor of Philosophy, 2018

Dissertation directed by: Professor Neil Spring
Department of Computer Science

Internet reliability for home users is increasingly important as a variety of services that we use migrate to the Internet. Yet, we lack authoritative measures of residential Internet reliability. Measuring reliability requires the detection of Internet outage events experienced by home users. But residential Internet outages are rare events. Further, they can affect relatively few users. Thus, detecting residential Internet outages requires broad and longitudinal measurements of individual users' Internet connections. However, such measurements of Internet reliability are challenging to obtain accurately and at scale.

Probing-based remote outage detection techniques can scale but their accuracy is questionable. These techniques detect Internet outages across time as well as across the IPv4 address space by sending active probes, such as pings and traceroutes, to users' IP addresses and use probe responses to infer Internet connectivity. However, they can infer false outages since their foundational assumption can sometimes be invalid: that the lack of response to an active probe

is indicative of failure. In this dissertation, I show how to use probing-based techniques to measure residential Internet reliability by defending the following thesis: *It is possible to remotely and accurately detect substantial outages experienced by any device with a stable public IP address that typically responds to active probes and use these outages to compare reliability across ISPs, media-types, geographical areas, and weather conditions*.

In the first part of the dissertation, I address the inaccuracy of probing-based techniques' detected outages and show how to use probe responses to correctly detect outages. I illustrate two scenarios where the lack of response to an active probe is *not* indicative of failure. In the first scenario, responses are delayed beyond the prober's timeout, leading these techniques to infer packet-loss instead of delay. In the second scenario, these techniques can falsely infer packet-loss when the address they are probing gets dynamically reassigned. I examine how often delayed responses and dynamic reassignment occur across ISPs to quantify the inaccuracy of these techniques. I show how outages can be inferred correctly even in networks with dynamic reassignment using complementary datasets that can reveal whether an address was dynamically reassigned before, during, and after a detected outage for that address.

In the second part of the dissertation, I motivate why the detection of individual addresses' outages is necessary for analyzing residential reliability. An individual address typically represents one residential customer; therefore, detecting outages for individual addresses can allow capturing even small outages. Prior probing-based techniques focus upon the detection of edge network outages affecting a substantial set of addresses belonging to a BGP prefix or to a /24 address block. Here, I quantitatively demonstrate the extent to which prior techniques can miss residential outages. I show that even individual address outages occur rarely in most networks. When multiple simultaneous outages of related individual addresses occur, there is likely a common underlying cause. With this insight, I develop and evaluate an approach to find outage events that are statistically unlikely to have occurred independently. I show that the majority of such events do not affect entire /24 address blocks or BGP prefixes, and are therefore not likely to be detected by existing techniques which look for outages at these granularities.

In the final part of the dissertation, I show how to use individual addresses' outages detected by probing-based techniques to assess Internet reliability across media-types, geographical areas, and weather conditions. Individual outages are not direct measures of reliability: they can occur independently because users disable equipment or can be observed falsely due to dynamic address renumbering. I use the insight that the statistical change in outage rate in different challenging environments (e.g., thunderstorm) can quantitatively expose actual outage "inflation". I show how to study the effect of challenging environments upon the reliability of a group of addresses by analyzing the inflation in outage rate for that group during its presence.

This dissertation's contributions will help achieve comprehensive measurements of Internet reliability that can be used to identify vulnerable networks and their challenges, inform which enhancements can help networks improve reliability, and evaluate the efficacy of deployed enhancements over time.

ANALYZING INTERNET RELIABILITY REMOTELY WITH PROBING-BASED TECHNIQUES

by

Ramakrishna Padmanabhan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Neil Spring, Chair/Advisor
Professor David Levin
Professor Bobby Bhattacharjee
Professor John Dickerson
Professor Mark Shayman

Acknowledgments

Throughout the time I have worked on this dissertation, I have had mentorship, help, and support, from many people. To all of them: Thank you very much. I am truly grateful.

I would first like to thank my advisor, Neil Spring. His mentorship has been invaluable in the making of this dissertation, and on a broader level, in the making of the researcher that I am today. His passion for research and attention to detail are qualities that I look up to and have learned to emulate. He has ever been a bastion of constructive criticism and his feedback has helped tremendously in honing my research skills. In addition, he has been flexible and accommodative of my occasionally unusual schedules. I could not have asked for more from an advisor.

There were several other mentors at the University of Maryland who had a direct role in shaping this dissertation. I have worked closely with Dave Levin and have benefitted much from his approach to research, his clear writing style, his brilliant presentations, his generosity, and his wonderful sense of humor. He has been a source of immense support and is an inspiration. Aaron Schulman helped me choose University of Maryland, introduced me to the problem I tackled in this dissertation, and importantly, also helped me sustain my research during difficult times. Aaron's enthusiasm, curiosity, and work-ethic, inspired me to persist with graduate school. I am also very grateful to Bobby Bhattacharjee; his incisive questions and feedback throughout have helped this dissertation significantly and have also helped me grow as a researcher. Ashok Agrawala provided valuable guidance early on and Atif Memon gave helpful comments during the thesis proposal. I would also like to thank my committee members, John Dickerson and Mark Shayman, for their feedback on the dissertation manuscript and presentation.

Mentors outside UMD have also provided valuable assistance during different stages of the dissertation. Alberto Dainotti has been instrumental during the finishing stages of the dissertation and has provided valuable feedback and support. Arthur Berger is another mentor who has helped considerably during the latter half of the dissertation; his thoughtfulness, ability to listen, and patience are qualities that I aim to emulate. I am also grateful to Matthew Luckie, Amogh Dhamdhere, and kc claffy, for introducing me to CAIDA and for helping me navigate the academic research world outside UMD. I would also like to thank Dave Plonka for showing me how to approach research with infectious enthusiasm. A special thank you to Krishna Sivalingam for giving me the opportunity to pursue my first research problem, and for his considerable help with graduate school applications.

I am also very grateful to my colleagues and friends, who have made graduate school an enjoyable experience. Aaron, Randy, Lex, Yunus, Greg, Karla, and Jessy were brilliant colleagues who welcomed me to graduate school and helped me learn the ropes. Matt Lentz has been an extraordinary colleague and friend, helping me immensely with my research and presentations, while also being a participant in some of the more memorable conversations I've had. I am also very grateful to Zhihao Li for his friendship and support; I am especially privileged to have been his first co-author on a paper. I am also grateful to Philipp Richter for a delightful paper-writing experience together. James, Youndo, Stephen, Katura, Richie, thank you all for your support and encouragement, especially during the thesis proposal and dissertation defense. Thank you also for making the lab a great place to hang out. A special thank you to Brandi Adams for being a wonderful friend and for having shared in the journey. Sharron McElroy, thank you for the conversations and for making the reimbursement process enjoyable. My housemates and fellow PhD students, Bhaskar Ramasubramaniam, Amit Chavan, and Kartik Nayak: I am truly grateful for all the tea, food, help and support. Anshul Sawant, Manish Purohit, Meethu Malu, and Sudha Rao, thank you for all the memories over the years. And Kleoniki Vlachou, though your cakes were one of the highlights of my first couple of years of graduate school, it is your support and encouragement through the years that have been the true icing on the cake.

I am also grateful to the students I've had the privilege to mentor and teach. Patrick Owen was the first undergraduate student I worked with and I am gratified that he is continuing to work on similar topics. Working with Ramakrishnan Sundara Raman and Reethika Ramesh was truly a delight; their enthusiasm and eagerness to learn made for an excellent mentoring experience.

Last, but by no means the least, I would like to thank my family—including my grandparents, uncles, aunts, cousins, and even my nephews and nieces—for their unceasing love and encouragement. I am especially grateful to my parents, Padmanabhan Raman and Lalitha Ramakrishnan for teaching me values like dedication, diligence, and optimism; without these values, I couldn't have completed this dissertation. I am also immensely grateful to my parents for always giving me the freedom to pursue my dreams. Ranjani Padmanabhan, my cheerful and witty sister, has been ever ready with her support. My wife, Janani Saikumar, who has also been pursuing her PhD alongside me, is one of the principal reasons this dissertation was possible. From providing me with initial encouragement to pursue graduate school, to continuous love and support during the ebbs and flows of graduate life, she has played an immense part.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Residential Internet reliability is increasingly important as a variety of services that we use migrate to the Internet. Internet users today can communicate with each other, perform financial transactions, plan their travel, and even obtain critical services such as health monitoring [1, 2] and emergency services [3, 4] from their homes. Our dependence upon the Internet is rising further as more of our home devices become connected. Consequently, reliable residential Internet connectivity is critical.

Broad and longitudinal measurements of users' Internet reliability can identify vulnerable networks, these networks' challenges, and potential enhancements. For instance, weather conditions such as thunderstorms, rain, and gales, can adversely affect Internet reliability [5]. Measurements can inform which areas are particularly vulnerable to weather conditions. Comparing measurements against other areas with similar weather conditions can provide insights on potential enhancements: for example, areas may be less vulnerable to gales where Internet cables run underground. Once an enhancement is deployed, measurements can reveal if the enhancement has resulted in improved Internet reliability.

The inferences from residential Internet reliability measurements can benefit various stakeholders, including policymakers, Internet Service Providers (ISPs), and residential users themselves. Policymakers around the world have begun efforts to measure Internet reliability [6–9], since such measurements can drive incentives and policies to improve reliability. ISPs can benefit from these measurements in multiple ways. Since even large ISPs rely upon their users to inform them of network connectivity issues [10], they may be unaware of problems in their own networks; these measurements can help ISPs recognize underyling problems. Further, ISPs can learn about the reliability of their competitors. Measurements of Internet reliability will also benefit residential users, since they can take into account the reliability of ISPs in their geographic region when purchasing Internet services.

However, we currently lack authoritative measures of residential Internet reliability, due to several challenges associated with obtaining such measures.

1.1 Background: state of the art in measuring residential Internet reliability

Intuitively, a reliable Internet connection is one that works continuously. In other words, it experiences no outages.

Measuring Internet reliability, therefore, necessitates measuring *Internet outages* and then using measured outages in a metric that represents some property of outages. At a high level, an Internet outage is an event that prevents users from communicating over the Internet. Since we expect outage events to be rare, detecting them requires monitoring a broad set of residential users over long periods. After detecting outages over time for a group of residential addresses—say addresses belonging to the same ISP or geographic region—the detected outages can be used to calculate reliability metrics. An example metric is the *outage rate*, the number of outages occurring over time for a group of addresses. These metrics can be used to compare different groups of addresses and to identify groups with particularly low reliability.

1.1.1 Challenges

Detecting residential Internet outages is challenging. The first challenge is the scale of residential users on the Internet: there are millions of residential Internet connections to monitor. Further, residential Internet outages can vary in the number of affected users. They can affect entire cities during large power outages. They can also affect just an individual house if a fallen tree branch damages the last-mile link between the house and the ISP. Another challenge is the heterogeneity of residential Internet connections. Some connections are cable connections, where the home router typically has a stable public IPv4 address. Others are DSL connections where the address assigned to the home router can change every 24 hours. Residential Internet connections can also use satellite links, which are prone to higher latencies.

Designing an outage detection system that can measure users broadly and yet remain accurate across diverse heterogeneous Internet connections remains a challenge.

1.1.2 Prior approaches

Prior approaches tradeoff either outage detection breadth or accuracy.

Edge network outage detection techniques detect outages broadly but focus upon detecting outages that affect a substanial numer of addresses in a group collectively. The group may comprise addresses belonging to the same /24 address block [11, 12], BGP prefix [13], or country [14]. Techniques seek such disruption events because individually, each large disruption has impact and their size makes them easier to confirm, e.g., with operators. However, residential Internet outages may be limited to a small neighborhood or apartment block; these techniques are likely to miss such events. Thus, they trade off outage detection accuracy for breadth.

On-premises techniques, such as RIPE Atlas [15], SamKnows [16], BISmark [17], and DIMES [18] measure diverse aspects of users' Internet connections, including connectivity problems, but measure relatively few users. These techniques deploy dedicated hardware or software at user premises that continuously conduct ping, traceroute, DNS measurements etc.; some of these measurements can be used to infer Internet connectivity problems.

Whereas on-premises techniques have fundamental scaling difficulties owing to manufacturing and deployment costs, hundreds of millions of IP addresses respond to active probes [19]. Since many residences have at least one device with a public IP address [20] (typically the home router), these IP addresses can be probed remotely, from vantage points under researcher control, to measure their connectivity. Thunderping [5] and Trinocular [11] adopt this approach to outage detection: they focus upon measuring only connectivity but do so for many users. Since these techniques can send probes remotely from servers under their control, without requiring any user involvement, they are able to detect outages across time as well as across the IPv4 address space.

Though capable of measuring Internet outages broadly, probing-based remote outage detection techniques can make false inferences about outages when some scenarios occur [19, 21]. The likelihood of occurrence of these scenarios varies across geographic regions, ISPs, and media type. Analyzing outages in the presence of these confounding factors requires broad measurements of these factors in turn.

1.2 Thesis

The goal of this dissertation is to provide broad, longitudinal, and accurate measurements of Internet reliability across ISPs, media-types, and geographic locations in a variety of circumstances. I work towards this goal using the probing-based technique due to its ability to scale. In the rest of the dissertation, I illustrate my approaches to mitigate probing-based techniques' problems in measuring residential Internet reliability by defending the following thesis:

It is possible to remotely and accurately detect substantial outages experienced by any device with a stable public IP address that typically responds to active probes and use these outages to compare reliability across ISPs, media-types, geographical areas, and weather conditions.

- *Device with a stable public IP address*: This is a device connected to the Internet, like a home-router, to which an ISP has assigned a public IP address such that the assignment is either static, or dynamic in a manner that allows the duration of dynamic assignment to be estimated.
- *Substantial outage*: I define a substantial outage to be an event where a device with an Internet connection is unable to send or receive any packets for at least 11 minutes. Such outages are likely to inconvenience residential users.
- *Accuracy of outage detection*: An outage detection technique is accurate when it correctly identifies every substantial outage event experienced by an Internet-connected-device. There are no time-periods when the address experiences a substantial outage but it goes undetected (false negatives). Similarly, there are no time-periods classified as outages when the destination address is able to receive packets from the Internet (false positives).
- *Reliability*: I measure reliability using the outage rate metric, which I define as the raw count of outage events over measured time.

1.3 Contributions

To demonstrate the thesis, I measure two confounding factors—high latency (Chapter 3) and dynamic address reassignent (Chapter 4)—that can lead probing-based outage detection techniques to make false outage inferences. In Chapter 5, I motivate the detection of individual addresses' outages. I go on to show how to measure Internet reliability in the presence of inference errors and unrelated outages in Chapter 6. This dissertation is organized as follows:

Chapter 2: State of the art in residential Internet outage detection

I provide background and place existing work in Internet outage detection in context. I describe the challenges that probing-based remote outage detection techniques will need to address to measure residential Internet reliability. These techniques study outages by sending active probes (such as ping's echo requests) and use probe responses to infer outages. They assume that a response to an active probe indicates a working path to the probed user device and that lack of response is indicative of failure. I illustrate two scenarios where this assumption can be invalid, leading to potentially false outage inferences.

Chapter 3: Mitigating false inferences due to early timeout

I investigate the prevalence of delayed probe responses due to early timeout. The lack of response to an active probe isn't always indicative of loss; for example, when responses are delayed beyond the prober's timeout, the response eventually arrives but the prober would never see the response because it timed out too early. I report how commonly responses are delayed beyond timeouts in networks around the world and use these measurements to discuss techniques that would mitigate this problem.

Chapter 4: Mitigating false inferences due to dynamic addressing

I investigate how dynamic addressing can lead remote probing-based outage detection techniques to make false inferences about outages and techniques to mitigate these false inferences. I measure the frequency and patterns in dynamic address reassignment for ISPs across the world. I also introduce a technique using a complementary dataset to determine whether an outage detected for an address by a probing-based system is a false outage due to dynamic reassignment.

Chapter 5: The need for measuring individual address outages

I motivate the need to study individual address outages by showing that individual address outage measurements can be used to find outage events that are statistically unlikely to have occurred independently, and that many of these events would not be detected by prior work. I describe how to use simultaneous outages of individual addresses related to each other, by geography and ISP, to identify outages that are highly unlikely to have occurred independently, and are therefore likely to share a common underlying cause.

Chapter 6: Analyzing weather's effect on Internet Reliability

I show how to measure and compare the reliability of groups of addresses—like addresses belonging to the same ISP, media-type, geographic region—when facing challenging environments. I consider one class of challenging environments that residential Internet connections can face: severe weather conditions.

I show how to use the inflation in outage rate to measure the effect of different classes of weather upon various groups of addresses.

Chapter 7: Future Work

I describe directions for future work in measuring residential reliability using probing-based techniques.

Chapter 2: State of the art in residential Internet outage detection

In this chapter, I begin with an overview of Internet outage measurement with a focus upon residential outage measurement. Next, I discuss probing-based techniques to detect outages remotely in detail and show their potential to measure residential users at scale. Then I illustrate scenarios where these techniques could make false inferences about outages.

2.1 Outage detection: an overview

The architects of the Internet predicted that network outages could occur, and designed the Internet to have the ability to route around outages [22]. As predicted, a variety of factors cause outages in the Internet, including optical fiber cuts [23], routing and infrastructure failures [24, 25], and hurricanes [5].

Large Internet outages that can affect packets from thousands of Internet hosts have received attention from the research community [11, 13, 13, 26–36]. Outages occurring in the Internet's core can cause Internet path failures; researchers have investigated transient Internet path failures caused by route changes [33–36] and longer path failures caused by infrastructure device outages [13, 27–32].

Dainotti et al. [14] observe Internet Background Radiation traffic sent to IPv4 darknets to detect outages affecting entire countries.

Another class of techniques detects outages at the Internet's edge, for network prefixes or address blocks, but does not target outages of individual users' Internet connections. Hubble studies reachability problems affecting BGP prefixes [13]. Trinocular detects outages affecting /24 address blocks. Richter et al. [12] use the observation point of a large CDN to detect periods of reduced activity from /24 address blocks consistent with outages. CAIDA's IODA system [37] detects outages affecting countries, ASNs, and geographic provinces using three complementary datasets: BGP updates from Routviews [38] and RIPE RIS [39], active probing data obtained with CAIDA's implementation of the Trinocular methodology, and IBR data using the technique introduced by Dainotti et al. [14].

However, outages that affect individual users have received comparatively less attention [5, 40–42]. In the rest of this chapter, I classify these efforts to detect outages into on-premises outage detection techniques and remote probing-based outage detection techniques, and discuss their approaches and challenges.

2.2 On-premises outage detection techniques

Recognizing the need for long term measurements of residential Internet performance, policymakers such as the FCC from the U.S., and Ofcom from the U.K. have deployed the SamKnows hardware platform [16] inside residences to measure residential Internet connections continuously by performing active and passive measurements and reporting their results to users, ISPs, and policy makers. RIPE NCC, the European RIR, has pioneered the RIPE Atlas [15] project and Sundaresan et al. the BISmark project [17], which also study user connectivity using dedicated hardware measurement devices on user premises. On-premises techniques can also use measurements from software deployed on user machines: the DIMES project [18] and DASU are two notable examples [43].

Hardware-based approaches can offer accurate reports about Internet connectivity since the hardware devices are designed to make measurements continuously as long as they are powered. These techniques have the ability to perform a range of other measurements such as DNS anycast tests that can identify which instance of a root-server is closest, and even passive measurement of the websites that users access. However, these approaches are fundamentally limited in scale since their hardware is expensive, distributing the hardware to users is time consuming, and convincing users to keep their hardware running is challenging. For example, the RIPE Atlas project, which began in 2010 and has been continuously expanding across the world, has fewer than 10,000 probes that are currently making measurements, out of more than 15,000 distributed probes.

While some of these costs can be offset by utilizing measurements from deployed software on user systems [18, 43, 44] or using a combination of hardware and software measurements [45], deploying software widely remains challenging. Separating user behavior, such as turning off their laptops, from Internet outage events presents additional challenges for these techniques [44].

2.3 Probing-based remote outage detection techniques

Probing-based remote outage detection techniques can detect connectivity problems remotely through active probing from servers under reseacher control. Though this approach will prevent certain types of measurements, such as DNS anycast tests, it can measure Internet connectivity for individual users at scale. However, existing techniques can infer false outages in some scenarios as I illustrate next.

Probing-based remote outage detection techniques study connectivity problems by sending active probes (such as ping's echo requests) and use probe responses to infer connectivity problems. For example, an echo-response from the end-host indicates that its network connection is working. If a previously responsive destination ceases to respond to probes, current techniques infer that the destination could be experiencing connectivity problems. Thunderping [5], Trinocular [11], and Zmap [46], have used this technique to detect outages, albeit at different scales. I discuss each approach in detail next.

2.3.1 Trinocular detects failures of /24 address blocks

Trinocular pings addresses in 4M /24 address blocks and uses the responses to detect Internet outages affecting entire blocks. Using historical data from the ISI census [47], it models the responsiveness of blocks and finds subsets of addresses within each block that are likely to respond to pings. The system pings a few of these addresses from each block at random and probes them in 11-minute rounds. Trinocular then employs Bayesian inference to reason about responses from blocks. When a block's responsiveness is lower than expected, Trinocular probes the block at a faster rate and eventually detects an outage when the follow-up probes also indicate the block's lack of Internet connectivity.

2.3.2 Thunderping detects failures of individual addresses during severe weather

Thunderping pings sampled addresses from multiple ISPs in geographic areas in the United States. Originally designed to evaluate how weather affects Internet outages, the system uses Planetlab vantage points to ping 100 IPv4 addresses from multiple ISPs in U.S. counties with active weather alerts. Each address is pinged from multiple Planetlab vantage points (at least 3) every 11 minutes, and addresses in a county are pinged six hours before, during, and after a weather alert for that county.

2.3.3 Zmap was used to study Internet outages during Hurricane Sandy

Zmap is an active probing technique designed to send packets of a specified type (such as ICMP echo) to all IPv4 addresses at very fast speeds (under an hour in 2013 [46], under 5 minutes today [48]. A key to these speeds is that the Zmap tool chooses to not store state at the prober; instead, response packets are matched with sent ones by encoding destination-specific data in the sent packets. By using cyclic generators, Zmap probes destination addresses in a random order, reducing probing burden on individual ISPs. However, Zmap's decision to not store state comes with a trade off: probe retransmissions upon the detection of a lost probe is difficult. The Zmap tool was used to detect Internet outages during Hurricane Sandy [46].

2.4 Probing-based techniques can scale but require improved accuracy

Since probing-based techniques send probes from machines under reseacher control, they have control over the number of addresses they probe and how frequently to probe. The Zmap technique has demonstrated that it is possible to send a ping to the entire IPv4 address space in less than five minutes [48].

However, probing-based remote outage detection techniques can infer false outages as a consequence of their foundational assumption: that a response to an active probe indicates a working path to the probed IP address and that lack of response is indicative of failure. Current techniques can make false positive inferences about outages in the following scenarios:

2.4.1 Confusing delay with loss

Traditionally, active probe based approaches time out probes after a few seconds. Thunderping [5] and Trinocular [11] time out probes after a few seconds. Responses that arrive after the timeout will be reported as lost. When this happens, existing techniques would infer loss though the responses are in fact merely delayed. Chapter 3 presents a measurement study on probe response latencies in networks around the world and discusses approaches to disambiguate delayed probes from lost probes.

2.4.2 Making false inferences about outages due to dynamic addressing

Consider an IP address that was previously responsive. If the host to which that IP address was assigned changed its IP address as a result of dynamic addressing, and if the probed IP address is not reassigned to any host, then echo responses will cease to arrive. Existing techniques would thus infer false probe-loss and consequently, false outages. Consider an alternate scenario where the probed IP address has an outage. Suppose that at some point during the outage, the IP address is reassigned to some other end-host which responds to probes. Existing techniques would infer that the arrival of responses signals the end of the outage and would infer that the outage ended prematurely. I address how to mitigate false inferences due to dynamic address reassignment in Chapter 4.

Chapter 3: Mitigating false inferences due to early timeout

In this chapter, I begin by describing how probe responses delayed beyond timeouts used by current probing-based techniques can lead to false probe-loss inferences, and thereby to false outage inferences.

Next, I describe work with colleagues that measured how frequently responses to active probes are delayed beyond timeouts set by existing approaches. We began by studying ping latencies from Internet-wide surveys [47] conducted by ISI, including 9.64 billion ICMP Echo Responses from 4 million different IP addresses in 2015, and identified addresses that are particularly likely to be subject to high delay. We then *verified* the high latencies by repeating measurements using other probing techniques, comparing the statistics of various surveys, and investigating high-latency behavior of ICMP compared to UDP and TCP. Finally, we explained these distributions by isolating satellite links, considering sequences of latencies at a higher sampling rate, and classifying a complete sample of the Internet address space through a modified Zmap client. These results are reproduced from our published work [19].

Using these results, I discuss how probing-based outage detection techniques can mitigate false outage inferences caused by delayed responses.

3.1 Challenges in selecting a timeout for probing techniques

Conventional wisdom suggests that active probes on the Internet should timeout after a few seconds. The belief is that after a few seconds there is a very small chance that a probe and response will still exist in the network. Once a probe times out, the prober can free the state associated with the probe, thereby reclaiming memory.

Conventional wisdom also suggests that a single timed out probe is insufficient to reason about end-host failures, due to potential random loss on the Internet. For most probing systems, any timed out active probes are followed up with retransmissions to increase the confidence that a lack of response is due to an outage event and not due to random loss on the Internet. These followup probes will also have a timeout that is generally the same as the first attempt.

Setting correct timeouts is critical for probing-based remote outage detection techniques. These techniques infer outages based upon lost probes and probe response loss is dependent upon the prober's timeout. Additonally, since probe timeouts trigger followup probes, setting appropriate timeouts is vital to these techniques.

However, choosing an appropriate timeout is challenging. Selecting a timeout value that is too low will ignore delayed responses and might add to congestion by performing retransmissions to an already congested host. Timeout values that are too high will delay retransmissions that can confirm an outage. In addition, too-high timeouts increase the amount of state that needs to be maintained at a prober, since every probe will need to be stored until either the probe times out, or the response arrives.

3.1.1 Timeouts used in outage and connectivity studies

Outage detection systems such as Trinocular [11] and Thunderping [5] tend to use a 3 second timeout for active probes because it is the default TCP SYN/ACK timeout [49]. Both techniques will not infer outages if a single response is delayed beyond the timeout, since they send follow-up probes to confirm suspected outages. However, if a series of responses are delayed beyond the timeout, both techniques can potentially infer false probe-loss and therefore, false outages.

Internet performance monitoring systems use a wide range of probe timeouts. On the shorter side, iPlane [50] and Hubble [13] send ICMP echo requests with a 2 second timeout. iPlane declares a host unresponsive after one failed retransmission. Hubble waits two minutes after a failed probe then retransmits probes six times and finally declares reachability with traceroutes. On the longer side, Feamster et al. [51] used a one hour timeout after each probe. However, they chose a long timeout to avoid errors due to clock drift between their probing and probed hosts; they did not do so to account for links that have excessive delays. PlanetSeer [52] assumed that four consecutive TCP timeouts (3.2-16 seconds) indicates a path anomaly.

It is especially important for connectivity measurements from probing hardware placed inside networks to have timeouts because of the limited memory in the probing hardware. The RIPE Atlas [15] probing hardware sends continuous pings to various hosts on the Internet to observe connectivity. The timeout for their ICMP echo requests is 1 second [53]. The SamKnows probing hardware uses a 3 second timeout for ICMP echo requests sent during loaded intervals [16].

We started this study with the expectation that these timeout values might need minor adjustment to account for large buffers in times of congestion; what we found was quite different.

3.2 Primary dataset overview

In this section, we describe the ISI survey dataset we use for our analysis of ping latency. We perform a preliminary analysis of ping latency and find that the dataset contains different types of responses that should (or should not) be matched to identify high-latency responses. Finally, we describe techniques to remove responses that could induce errors in the latency analysis.

3.2.1 Raw ISI survey data

ISI has conducted Internet wide surveys [47] since 2006. Precise details can be found in Heidemann et al. [47], and technical details of the data format online [54], but we present a brief overview here.

Each survey includes pings sent to approximately 24,000 /24 address blocks, meant to represent 1% of all allocated IPv4 address space. Once an address block is included, ICMP echo request probes are sent to all 256 addresses in the selected

/24 address blocks once every 11 minutes, typically for two weeks. The blocks included in each survey consist of four classes, including blocks that were chosen in 2006 and probed ever since, as well as samples of blocks that were responsive in the last census—another ISI project that probes the entire address space, but less frequently. However, we treat the union of these classes together.

We use data from 103 surveys taken between April 2006 and February 2015, and performed initial studies based on 2011–2013 data, but focus on the most recent of them, in January and February of 2015 for data quality and timeliness. The dataset consists of all echo requests that were sent as part of the surveys in this period, as well as all echo responses that were received. Of particular importance is that echo responses received within, typically, three seconds of an echo request to the same address are matched into a single record and given a round-trip measurement precise to microseconds. Should an echo response take four seconds to arrive, a "timeout" record is recorded associated with the probe, and an "unmatched" record is recorded associated with the response. These two packets have timestamps precise only to seconds. The dataset also includes ICMP error responses (e.g., "host unreachable"); we ignore all probes associated with such responses since the latency of ICMP error responses is not relevant.

In later sections, we will complement this dataset with results from Zmap [46] and additional experiments including more frequent probing with Scamper [55] and Scriptroute [56].

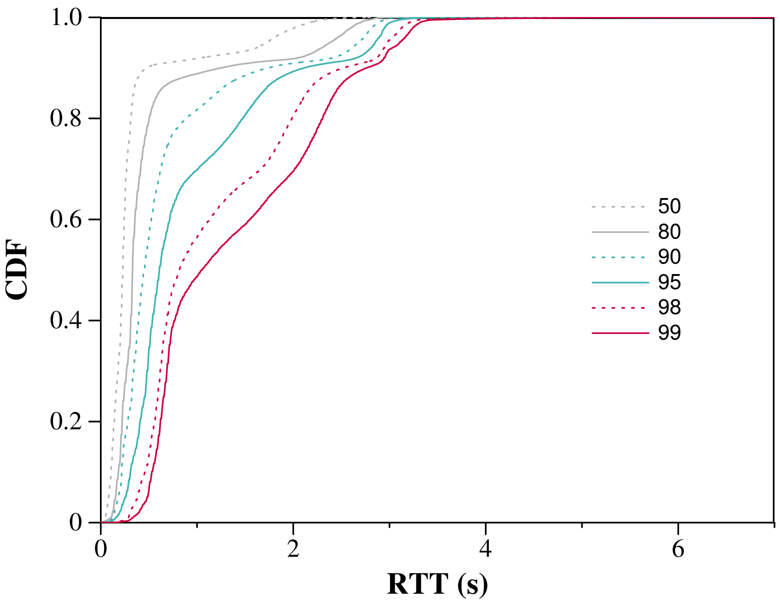


Figure 3.1: CDF of percentile latency of survey-detected responses per IP address: Each point represents an IP address and each curve represents the percentile from that IP address's response latencies. The slope of the latency percentiles increases around the 3 second mark, suggesting that ISI's prober timed out responses that arrived after 3 seconds.

3.2.2 Matched response latencies are capped at the timeout

In this section, we present the latencies we would observe when considering only those responses that were matched to requests because they arrived within the timeout. We call these responses *survey-detected responses*.

We aggregate round trip time measurements in terms of the distribution of latency values per IP address, focusing on characteristic values on the median, 80th, 90th, 95th, 98th and 99th percentile latencies. That is, we attempt to treat each IP address equally, rather than treat each ping measurement equally. This aggregation ensures that well-connected hosts that reply reliably are not over-represented relative to hosts that reply infrequently.

Taking ISI survey datasets from 2011–2013 together, we show a CDF of these percentile values considering only survey-detected responses in Figure 3.1. Taken literally, 95% of echo replies from 95% of addresses will arrive in less than 2.85 seconds. However, it is apparent that the distribution is clipped at the 3 second mark, although a few responses were matched even after 7 seconds.

We observe three broad phases in this graph: (1) the lower 40% of addresses show a reasonably tight distribution in which the 99th percentile stays close to the 98th; (2) the next 50% in which the median remains low but the higher percentiles increase; and (3) the top 10% where the median rises above 0.5 seconds.

3.2.3 Unmatched responses

If a probe takes more than three seconds to solicit a response, it appears as if the probe timed-out and the response was unsolicited or *unmatched*. Since it appears from Figure 3.1 that three seconds is short enough that it is altering the distribution of round trip times, we are interested in matching these echo responses to construct the complete distribution of round trip times.

Matching these responses to find *delayed responses* is not a simple matter, however. In particular, we find two causes of *unexpected responses* that should not yield samples of round trip times: unmatched responses solicited by echo requests sent to broadcast addresses and apparent denial of service responses.

We match a delayed response with its corresponding request as follows: Given an unmatched response having a source IP address, we look for the last request sent to that IP address. If the last request timed out and has not been matched, the latency is then the difference between the timestamp of the response and the timestamp of the request. ISI recorded the timestamp of unmatched responses to a 1 second precision, thus the latencies of inferred delayed responses are precise only to a second.

The presence of unexpected responses can lead to the inference of incorrect latencies for delayed responses using this technique: not all unexpected responses should be matched by source address. We thus develop filters to remove unexpected responses from the set of unmatched responses.

We note that it is possible to match responses to requests explicitly using the id and sequence numbers associated with ICMP echo requests, and even perhaps using the payload. These attributes were not recorded in the ISI dataset, which motivates us to develop the source address based scheme. We use these fields when running Zmap or other tools to confirm high latencies in Section 3.4 below.

3.2.3.1 Broadcast responses

The dataset contains several instances where a ping to a destination times out, but is closely followed by an unmatched response from a source address that is within the same /24 address block, but different from the destination. In each round of probing, this behavior repeats. Here, we analyze these unmatched responses, find that they are likely caused by probing broadcast addresses, and filter them.

Network prefixes often include a broadcast address, where one address within a subnet represents all devices connected to that prefix [57]. The broadcast address in a network should be an address that is unlikely to be assigned to a real host [57], such as the address whose host-part bits are all 1s or 0s, allowing us to characterize broadcast addresses. Devices that receive an echo request sent to the broadcast address may, depending on configuration, send a response [49], and if sending a response, will use a source address that is their own. We call these responses *broadcast responses*. No device should send an echo response with the source address that is the broadcast destination of the echo request.

We hypothesize that pings that trigger responses from different addresses within the same /24 address block result when the ping destination is a broadcast address. We examine ping destinations that solicit a response from a different address in the same /24 address block, and check if they appear to be broadcast addresses.

We extended the ICMP probing module in the Zmap scanner [46] to embed the destination into the echo request, then to extract the destination from the echo response. Doing so allows us to infer the destination address to which the probe was originally sent. Zmap collected the data and made it available for download at scans.io.

We choose the Zmap scan conducted closest in time to the last ISI survey we studied, on April 17 2015, to investigate the host-part bits of destination ad-

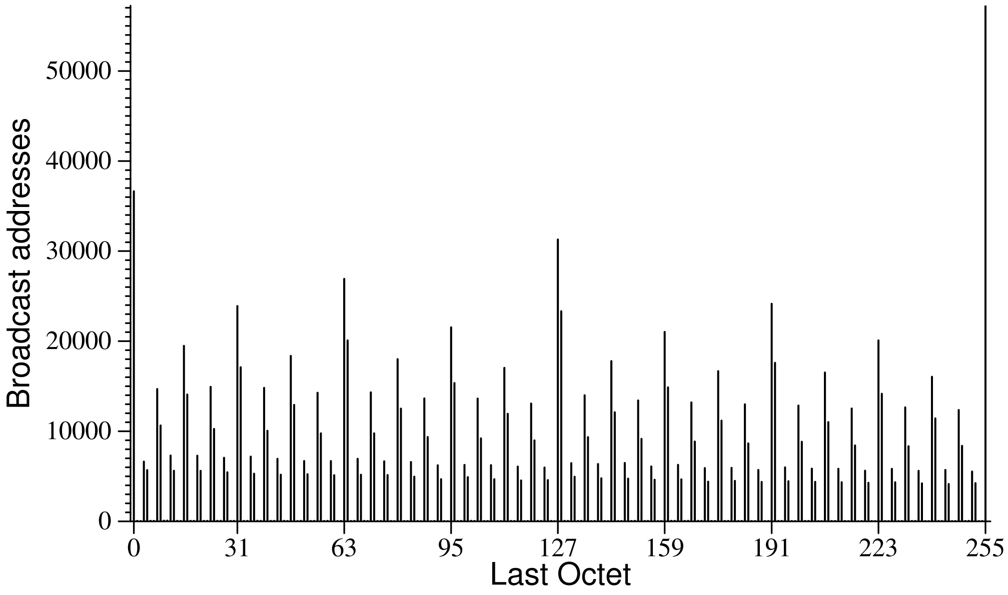


Figure 3.2: Broadcast addresses that solicit responses in Zmap: Broadcast addresses usually have last octets whose last N bits are either 1 or 0 (where N > 1).

dresses that triggered responses from a different address from the same /24 address block. We plot the distribution of the last octets of these addresses in Figure 3.2. Last octets with the last N bits ending in 1 or 0, where N is greater than 1, such as 255, 0, 127, 128 etc., have spikes. These addresses are likely broadcast addresses. On the other hand, last octets that end in binary ’01’ or ’10’ have very few addresses.

Broadcast responses exist in the dataset

We examine if unmatched responses in the ISI dataset are caused by pings sent to broadcast addresses. Since broadcast responses are likely to be seen after an Echo Request sent to a broadcast address, we find the most recently probed address

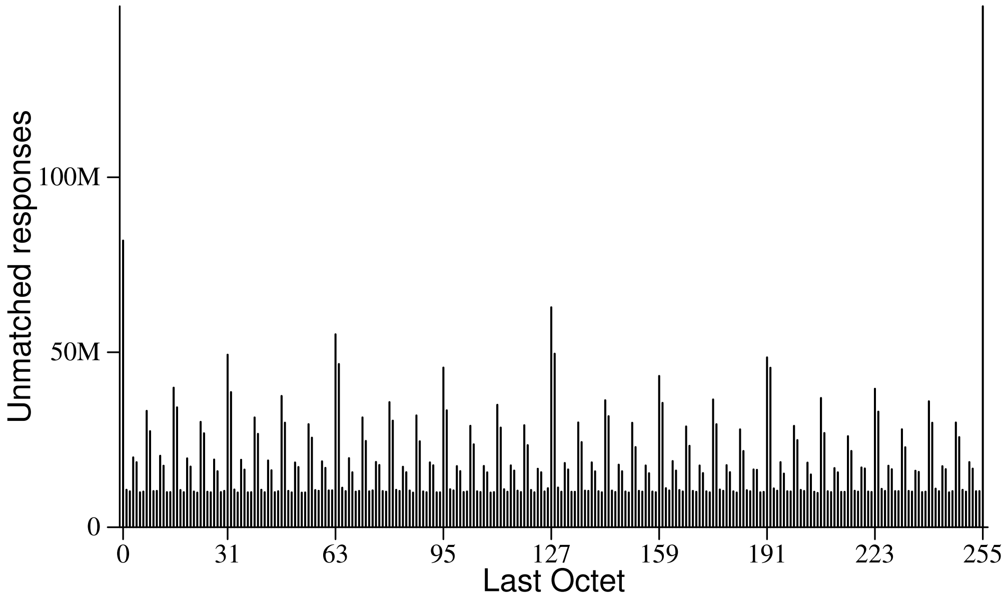


Figure 3.3: Broadcast addresses that solicit responses in ISI surveys: Number of unmatched responses that followed a probe sent to address with last octet X. Last octets with last N bits ending in 0s and 1s (where N > 1) observe spikes, likely caused by broadcast responses. Not all unmatched responses are caused by broadcast responses, however, since there exist roughly 10M unmatched responses distributed evenly across all last octets.

within the same /24 prefix for each unmatched response. We then extract the last octet of the most recently probed address. Figure 3.3 shows the distribution of unmatched responses across these last octets. We find that around 10M unmatched responses are distributed evenly across all last octets: these are unmatched responses that don't seem to be broadcast responses. However, last octets that have their last N bits as 1s and 0s ,when N is greater than 1, observe spikes similar to those in Figure 3.2.

If left in the data, broadcast responses could yield substantial latency overestimates in the following, common, scenario, which we illustrate in Figure 3.4. Assume that the echo request sent to an address 211.4.10.254 is lost and that the device is configured to respond to broadcast pings. The echo request sent to 211.4.10.254 could then be matched to the response to the request sent to 211.4.10.255, the broadcast address of the enclosing prefix. This would lead to a latency based on the interval between probing 211.4.10.254 and 211.4.10.255, as shown in the figure.

Filtering broadcast responses

We develop a method which uses ISI's non-random probing scheme to detect addresses that source broadcast responses. We call such addresses *broadcast responders*, and seek to filter all their responses. We believe that delayed responses are likely to exhibit high variance in their response latencies, since congestion varies over time. On the other hand, a broadcast response is likely to have relatively stable latency.

ISI's probing scheme sends probes to each address in a /24 address block in a nonrandom sequence, allowing us to develop a filter that checks if a source address responds to a broadcast address each round. Addresses are probed such that last octets that are off by one, such as 254 and 255, receive pings spaced 330 seconds apart (half the probing interval of 11 minutes) as shown in Figure 3.4. For every unmatched response with a latency of at least 10 seconds, the filter checks if

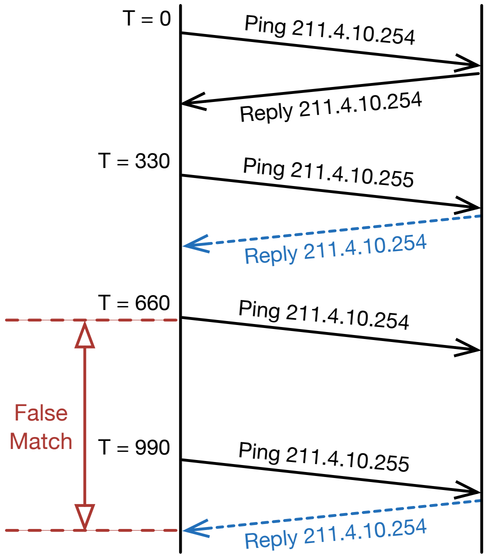


Figure 3.4: We filter broadcast responses since they can lead to the inference of false latencies. This figure illustrates a potential incorrect match caused by a broadcast response. Echo requests sent to the broadcast address 211.4.10.255 at T = 330 and T = 990 seconds solicit responses from 211.4.10.254. When a timeout occurs for a request sent directly to 211.4.10.254 at T = 660 seconds, we would falsely connect that request to the response at T = 990 seconds.

the same source address had sent an unmatched response with a similar latency in the previous round. We take an exponentially weighted moving average of the number of times this occurs for a given source address with $\alpha = 0.01$. Most broadcast responders have the maximum of this moving average $>$ 0.9, but since probe-loss can potentially decrease this value, we mark IP addresses with values $>$ 0.2 and filter all their responses.

We confirm that we find broadcast responders correctly in the ISI surveys by comparing the ones we found in the ISI 2015 surveys with broadcast responders from the Zmap dataset. Zmap detected 939,559 broadcast responders in the April 17 2015 scan, of which 7212 had been addresses that provided Echo Responses in ISI's IT63w (20150117) and IT63c (20150206) datasets. The filter detected 7044 (97.7%) of these as broadcast responders. We inspected the 168 remaining addresses and found that 154 addresses have 99th percentile latencies below 2.5 seconds. Since ISI probes a /24 prefix only once every 2.5 seconds, these addresses cannot be broadcast responders. Another 5 addresses have 99th percentiles latencies below 5 seconds; these are unlikely to be broadcast responders as well.

The remaining 9 addresses had 99th percentile latencies in excess of 300s and seem to be broadcast responders. Upon closer inspection, we found that these addresses only occasionally sent an unmatched response: around once every 50 rounds. The α parameter of the filter can tolerate some rounds with missing responses, but these addresses respond in so few rounds that they pass undetected. If these 9 are indeed broadcast responders as suggested by high 99th percentile latencies, this yields a false negative rate of our filter of 0.13%.

3.2.3.2 Duplicate responses

Packets can be duplicated. A duplicated packet will not affect inferred latencies as long as the original response to the original probe packet reaches the prober,

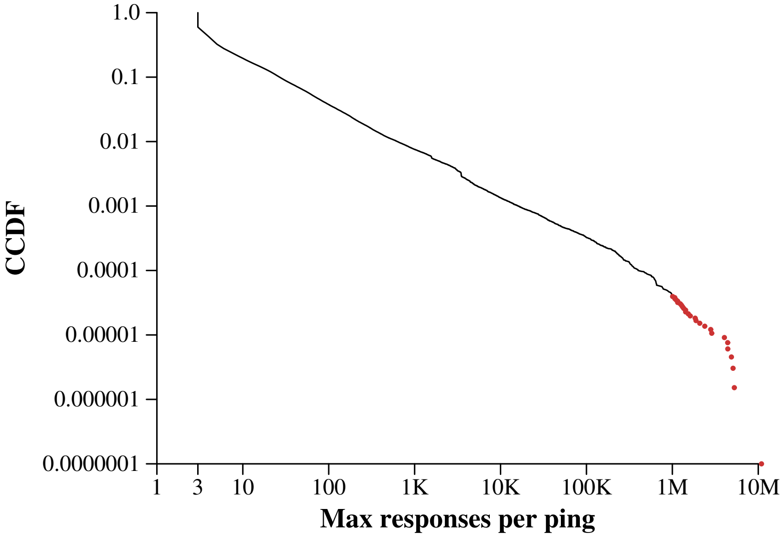


Figure 3.5: Maximum number of responses received for a single echo request, for IP addresses that sent more than 2 responses to an echo request. The red dots indicate instances where addresses responded to a single echo request with more than 1M echo responses. We believe that these are caused by DoS attacks.

since our scheme ignores subsequent duplicate responses. However, we find that some IP addresses respond many times to a single probe. In this case, the incoming packets aren't responses to probes, but are either caused by incorrect configurations or malicious behavior.

Figure 3.5 shows the distribution of the maximum number of echo responses observed in response to a single echo request. Since broadcast responses can also be interpreted as duplicate responses, we look only at IP addresses that sent more than 2 echo responses for an echo request. Of 658,841 such addresses, we find that 4,985 (0.7%) sent at least 1,000 echo responses. The red dots in the figure show

26 addresses that sent more than one million echo responses, with one address sending nearly 11 million responses in 11 minutes.

Zmap authors reported that they observed retaliatory DoS attacks in response to their Internet-wide probes [46]. We believe that some of the responses in the ISI dataset are also caused by DoS attacks.

We filter duplicate responses by ignoring IP addresses that ever responded more than 4 times to a single echo request, based on observing the distribution of duplicates shown in Figure 3.5. Packets can sometimes get duplicated on the Internet, and we want to be selective in our filtering to remove as little as necessary. Even if a response from the probed IP address is duplicated and a broadcast response is also duplicated, there should be only 4 echo responses in the dataset. We believe that IP addresses observing more than 4 echo responses to a single echo request are either misconfigured or are participating in a DoS attack. In either case, the latencies are not trustworthy.

3.3 Recommended Timeout Values

In this section, we analyze the ping latencies of all pings obtained from ISI's Internet survey datasets from 2015 to find reasonable timeout values. We demonstrate the effectiveness of our matching scheme for recovering delayed responses from the dataset. We then group the survey-detected responses and delayed responses together to determine what timeout values would be necessary to recover various percentiles of responses. Some IP addresses observe very high latencies in

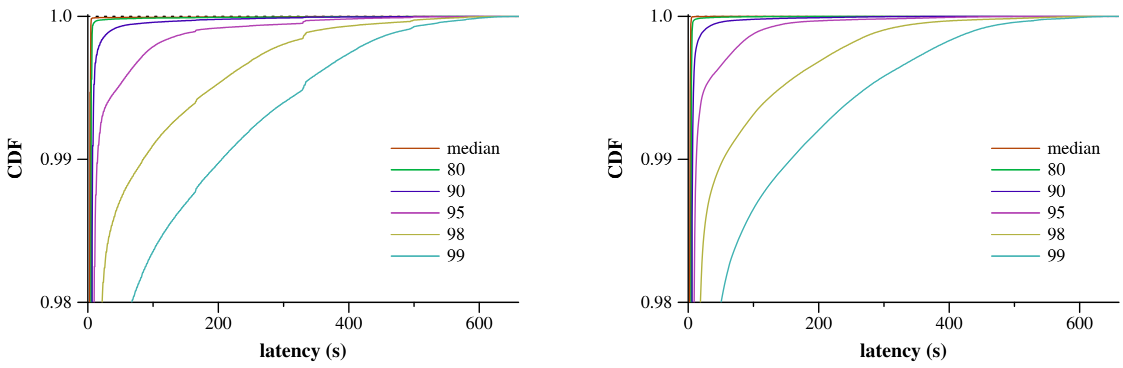
	Packets	**Addresses**
Survey-detected	9,644,670,150	4,008,703
Naive matching	9,768,703,324	4,008,830
Broadcast responses	33,775,148	9,942
Duplicate responses	67,183,853	20,736
Survey + Delayed	9,667,744,323	3,978,152

Table 3.1: Adding unmatched responses to survey-detected responses

the ISI dataset; we verify that these are real in Section 3.4 and examine causes in Section 3.5.

3.3.1 Incorporating unmatched responses

ISI detected 9.64 Billion echo responses from 4 Million IP addresses in 2015 in the IT63w (20150117) and IT63c (20150206) datasets, as shown in the first row of Table 3.1. The next row shows the number of responses we would have obtained if we had used a naive matching scheme where we simply matched each unmatched response for an IP address with the last echo request for that IP address, without filtering unexpected responses. The number of responses increases by 1.3% to 9.77 Billion; however, this includes responses from addresses that received broadcast responses and duplicate responses. After filtering unexpected responses, the number of IP addresses reduces to 99.23% of the original addresses. Of 30,678 discarded IP addresses, 9,942 (32.4%) addresses were dis-



(a) Before filtering

(b) After filtering

Figure 3.6: CDF of Percentile latency per IP address before and after filtering unexpected responses. Each point represents an IP address and each color represents the percentile from that IP address's response latencies. Before filtering unexpected responses, there are bumps caused by broadcast responses at 330s, 165s and 495s, fractions of the 11 minute (660s) probing interval.

carded because they also received broadcast responses. The majority of discarded IP addresses, 20,736 (67.6%) were addresses that sent more than 4 echo responses in response to a single echo request.

Though the number of discarded IP addresses is relatively small, removing them eliminates responses that cluster around 330, 165, and 495 seconds. Figure 3.6 shows the distribution of percentile latency per IP address before and after filtering unexpected responses. Comparing these two graphs shows that the "bumps" in the CDF are removed by the filtering.

After discarding addresses, our matching technique yields 23,074,173 additional responses for the remaining addresses, giving us a total of 9.67 Billion

		% of pings						
		1%	50%	80%	90%	95%	98%	99%
% of addresses	1%	0.01	0.03	0.04	0.07	0.10	0.13	0.18
	50%	0.16	0.19	0.21	0.26	0.42	0.53	0.64
	80%	0.19	0.26	0.33	0.43	0.54	0.74	1.21
	90%	0.22	0.31	0.42	0.57	0.84	1.61	3
	95%	0.25	1.42	2.38	3	5	9	15
	98%	0.30	1.94	4	6	12	41	78
	99%	0.33	2.31	4	8	22	76	145

Table 3.2: Minimum timeout in seconds that would have captured c% of pings from r% of IP addresses in the IT63w (20150117) and IT63c (20150206) datasets (where r is the row number and c is the column number).

Echo Responses from 3.98 Million IP addresses. We perform our latency analysis on this combined dataset.

3.3.2 Recommended Timeout Values

We now find retransmission thresholds which recover various percentiles of responses for the IP addresses from the combined dataset. For each IP address, we find the 1st, 50th, 80th, 90th, 95th, 98th and 99th percentile latencies. We then find the 1st, 50th, 80th, 90th, 95th, 98th and 99th percentiles of all the 1st percentile latencies. We repeat this for each percentile and show the results in Table 3.2.

The 1st percentile of an address's latency will be close to the ideal latency that its link can provide. We find that the 1st percentile latency is below 330ms for 99% of IP addresses: most addresses are capable of responding with low latency. Further, 50% of pings from 50% of the addresses have latencies below 190ms, showing that latencies tend to be low in general.

However, we see that a substantial fraction of IP addresses also have surprisingly high latencies. For instance, to capture 95% of pings from 95% addresses requires waiting 5 seconds. Restated, at least 5% of pings from 5% of addresses have latencies higher than 5 seconds. Thus, even setting a timeout as high as 5 seconds will infer a false loss rate of 5% for these addresses.

Note that retrying lost pings cannot be used as a substitute for setting a longer timeout since a retried ping is not an independent sample of latency. Whatever caused the first one to be delayed is likely to cause the followup pings to be delayed as well, as we show in Section 3.5.

At the extreme, we see 1% of pings from 1% of addresses having latency above 145 seconds! These latencies are so high that we investigate these addresses further. *We now consider 60 seconds to be a reasonable timeout to balance progress with response rate, at least when studying outages and latencies, although an ideal timeout may vary for different settings.* A timeout of 60 seconds easily covers 98% of pings to 98% of addresses, yet does not seem long enough to slow measurements unnecessarily.

3.4 Verification of long ping times

In this section, we address doubts that long observed ping times are real: that they are a product of ISI's probing scheme, that they might be caused by errors in a particular data set, or that they might derive from discrimination against ICMP.

3.4.1 Are high latencies observed by other probing schemes?

Some of the latencies in Table 3.2 are so high that we considered if they could be artifacts of ISI's probing scheme. We investigate latencies obtained using two other probing techniques, Zmap and scamper, and check if the high latencies observed in the ISI datasets are reproducible.

Does Zmap observe high latencies?

We check for high latencies using the Zmap scanner [46]. As part of our extension of the ICMP probing module in the Zmap scanner, we also embed the probe send time into the echo request, and extract it from the echo response, allowing us to estimate the RTT, albeit without the precision of kernel send timestamps.

Zmap has performed these scans since April 2015. Scans have been conducted over a range of different times, different days of the week and across four months in 2015 (as of Sep 5, 2015), as shown in Table 3.3. Typically, scans were performed on Sundays or Thursdays, beginning at noon UTC time. However, the scans on April 17, May 22, and June 15 were conducted on other days and

Scan Date	Day	Begin Time	Echo Responses
Apr 17, 2015	Fri	02:44	339M
Apr 19, 2015	Sun	12:07	340M
Apr 23, 2015	Thu	12:07	343M
Apr 26, 2015	Sun	12:07	343M
Apr 30, 2015	Thu	12:08	344M
May 3, 2015	Sun	12:08	344M
May 17, 2015	Sun	12:09	347M
May 22, 2015	Fri	00:57	371M
May 24, 2015	Sun	12:09	369M
May 31, 2015	Sun	12:09	362M
Jun 4, 2015	Thu	12:10	368M
Jun 15, 2015	Mon	13:53	357M
Jun 21, 2015	Sun	12:11	368M
Jul 2, 2015	Thu	12:00	369M
Jul 5, 2015	Sun	12:00	368M
Jul 9, 2015	Thu	12:00	369M
Jul 12, 2015	Sun	12:00	367M

Table 3.3: Zmap scan details: For each Zmap scan in Figure 3.7, the table shows the date, day of the week, the time at which the scan began (in UTC time), and the number of destinations that responded with Echo Responses.

at other times, increasing diversity. Each Zmap scan takes 10 and a half hours to complete and recovers Echo Responses from around 350M addresses.

We choose all available scans and analyze the distribution of RTTs for the Echo Responses in Figure 3.7. Most responses arrive with low latency, having a median latency lower than 250ms for each scan. However, 5% of addresses responded with RTTs greater than 1 second in each scan. Further, 0.1% of addresses responded with latencies exceeding 75 seconds in each scan although the 99.9th percentile latency exhibited some variation: the May 22 scan had the lowest 99.9th percentile latency (77 seconds) whereas the July 9 scan had the highest

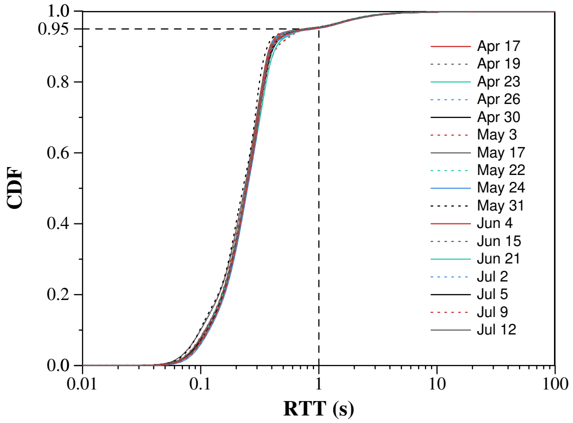


Figure 3.7: Distribution of RTTs for all Zmap scans performed in 2015. Around 5% of addresses have latencies greater than 1s in each scan, and 0.1% of addresses observed latencies in excess of 75s.

(102 seconds). We infer from these nearly identical latency distributions that high latencies are persistent for a consistent fraction of addresses.

Does scamper also observe high latencies?

Both ISI and Zmap probe millions of addresses, and we investigate whether latencies are affected by these probing schemes triggering rate-limits or firewalls. We select a small sample of addresses that are likely to have high latencies from the ISI dataset, probe them using scamper [55], and check for unusually high latencies.

In the 2011 - 2013 ISI dataset, 20,095 IP addresses had at least 5% of their pings with latencies 100 seconds and above. We chose 2000 random IP addresses from this subset and sent 1000 pings to them, once every 10 seconds using scamper [55] and analyzed the responses. In this analysis, we used scamper's default packet response matching mechanism: so long as scamper continues to run, received responses will be matched with sent packets. Because we used scamper's defaults, scamper ceased to run 2 seconds after the last packet was sent, so we missed responses to the last few pings that arrived after scamper ceased running. Although scamper can be configured to wait longer for responses, in later analyses, we ran tcpdump simultaneously and matched responses to sent packets separately.

Of the 2000 addresses, 1244 responded to our probes. Figure 3.8 shows the percentile latency per IP address. The 95th percentile latency for 50% of the addresses is now considerably lower, at 7.3s. This suggests that addresses prone to extremely high latencies vary with time: we investigate addresses with this behavior further in Section 3.5.

Nevertheless, Figure 3.8 shows that scamper also observes some instances of very high latencies. 17% of addresses observe latencies greater than 100 seconds for 1% of their pings. We therefore rule out the possibility that the high latencies are a product of the probing scheme.

3.4.2 Is it a particular survey or vantage point?

ISI survey data are collected from four vantage points at different times. Vantage points are identified by initial letter, and are in Marina del Rey, California, “w”; Ft. Collins, Colorado, “c”; Fujisawa-shi, Kanagawa, Japan, “j”; and Athens, Greece, “g”.

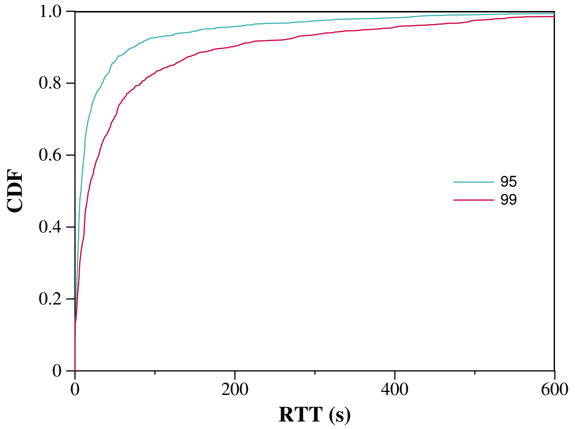


Figure 3.8: Confirmation of high latency: Percentile latency per IP address for 2000 randomly chosen IP addresses from ISI's 2011 - 2013 surveys that had $>$ 5% of pings with latencies 100s and above. Each point represents an IP address and the lines represent the percentile latency from that IP address. 17% of them continue to observe 1% of their pings with latencies $>$ 100s.

In this section, we look at summary metrics of each of the surveys. In Figure 3.9, our intent was to ensure that the results were consistent from one survey to the next, but we found a surprising result as well. The consistency of values is apparent: the median ping from the median address remains near 200ms for the duration. However, there are exceptions in the following data sets: IT59j (20140515), IT60j (20140723), IT61j (20141002), IT62g (20141210). These higher sampled latencies are coincident with a substantial reduction in the fraction of responses that are matched: in typical ISI surveys, 20% of pings receive a response; in these, between 0.02% and 0.2% see a response. It appears that these data sets should not be considered further. Additionally, it54c (20130524) it54j (20130618)

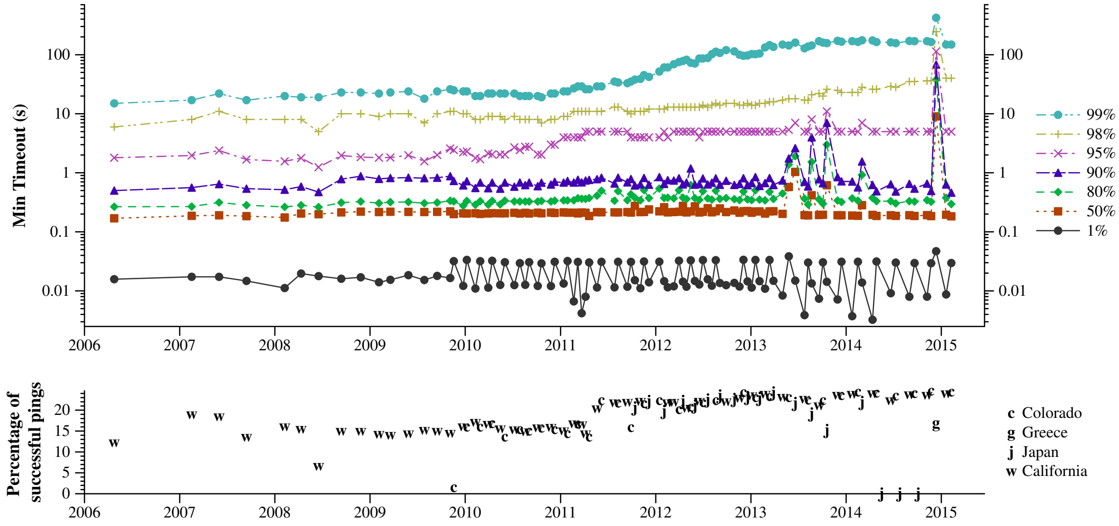


Figure 3.9: Top: Minimum timeout required to capture the c^{th} percentile latency sample from the c^{th} percentile address in each survey, organized by time. Each point represents the timeout required to capture, e.g., 95% of the responses from 95% of the addresses. The 1% line is indicative of the minimum. Bottom: Response rate for each survey; symbols represent which vantage point was used. Surveys from Japan with very few successes are not plotted on the top graph.

and it54w (20130430) were flagged by ISI as having high latency variation due to a software error [58].

Ignoring the outliers, trends are apparent. The timeout necessary to capture 95% of responses from 95% of addresses increased from near two seconds in 2007 to near five seconds in 2011. (We note that the apparent stability of this line may be misleading; since the y-axis is a log scale and our latency estimates are only precise to integer seconds when greater than 3, small variations will be lost.) The 98th percentile latency from the 98th percentile address has increased steadily since 2011, and the 99th increased from a modest 20 seconds in 2011 to a surprising 140 in 2013. These latency observations are not isolated to individual traces.

In sum, high latency is increasing, and although some surveys show atypical statistics, early 2015 datasets that we focus on appear typical of expected performance.

3.4.3 Is it ICMP?

One might expect that high latencies could be a result of preferential treatment against ICMP. RFC 1812 allows routers responding to ICMP to rate-limit replies [59, 60], however, this limitation of ICMP should not substantially affect the results since each address is meant to receive a ping from ISI once every eleven minutes. Nevertheless, one can imagine firewalls or similar devices that would interfere specifically with ICMP.

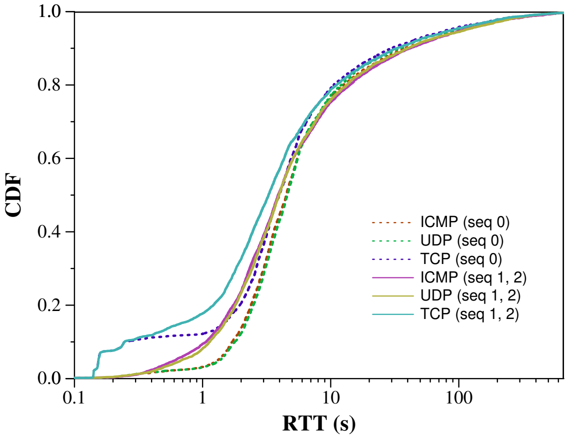


Figure 3.10: 98th percentile RTTs associated with high-latency IP addresses using different probe protocols. The first probe of a triplet (seq 0) often has a higher latency than the rest; TCP probes appear to have a similar distribution except for firewall-sourced responses.

To evaluate this possibility, we selected high-latency addresses from the IT63c (20150206) survey. To these addresses we sent a probe stream consisting of three ICMP echo requests separated by one second, then 20 minutes later, three UDP messages separated by one second, then again 20 minutes later, three TCP ACK probes separated by one second. We avoided TCP SYNs because they may appear to be associated with security vulnerability scanning. We then consider the characteristics of these hosts in terms of the difference between ICMP delay and TCP or UDP delay.

“High-latency” addresses to sample

We choose the top 5% of addresses when sorting by each of the median, 80th, 90th and 95th percentile latencies. Many of these sets of addresses overlap: those who have among the highest medians are also likely to be among the highest 80th percentiles. However, we considered these different sets to be important so that the comparison would include both hosts with high persistent latency and those with high occasional latency. After sampling 15,000 addresses from each of these four sets, then removing duplicates, we obtain 53,875 addresses to probe.

From these addresses, we found that only 5,219 responded to all probes from all protocols on April 29, 2015. This is somewhat expected: Only 27,579 responded to any probe from any protocol.

To complete the probing, we use Scamper [55] to send the probe stream to each of the candidate addresses. Note that scamper uses a 2s timeout by default although the timeout can be configured. Instead of setting an alternate timeout in Scamper, we run tcpdump to collect all received packets, effectively creating an “indefinite” timeout. This allows us to observe packets that arrive arbitrarily late since we continue to run tcpdump days after the Scamper code finished.

All protocols are treated the same (mostly)

For each protocol, we select the 98th percentile RTT per address and plot the distribution in Figure 3.10. We noticed two obvious features of the data: that the first packet of the triplet often had a noticeably different distribution of round trip times, and that the TCP responses often had a mode around 200ms. We will investigate the "first ping" problem in Section 3.5.3.

The TCP responses appear to be generated by firewalls that recognize that the acknowledgment is not part of a connection and sent a RST without notifying the actual destination: this cluster of responses all had the same TTL and applied to all probes to entire /24 blocks. That is, for each address that had such a response, all other addresses in that /24 had the same.

Ignoring the quick TCP responses apparently from a firewall, it does not appear that any protocol has significant preferential treatment among the high-latency hosts. Of course, this observation does not show that prioritization does not occur along any of these paths; our assertion is only that such prioritization, if it exists, is not a source of the substantial latencies we observe.

3.4.4 Summary

In this section, we confirmed that extremely high latencies are also observed by techniques besides ISI's. We find that the high latencies are not a result of a few individual ISI datasets, even though some did appear atypical. Further, high latencies affect all protocols the same.

We also found that the prevalence of high latencies has been increasing since 2011. In 2015, a consistent 5% of addresses have latencies greater than a second.

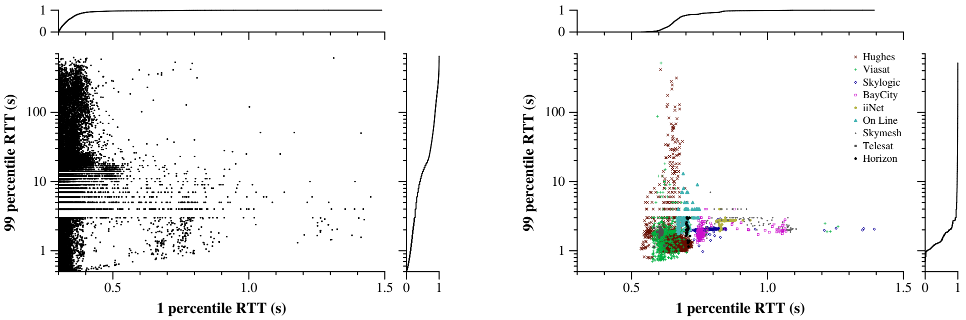


Figure 3.11: Scatterplot of 1st and 99th percentile latencies for addresses with high values of both in survey IT63c; Left omits satellite-only ISPs; Right includes only satellite-only ISPs.

3.5 Why do pings take so long?

In this section, we aim to determine what causes high RTTs. We investigate the RTTs of satellite links and find that they account for a small fraction of high RTT addresses. We follow up with an analysis of Autonomous Systems and geographic locations that are most prone to two potentially different types of high RTTs: RTTs greater than 1s and RTTs greater than 100s. We then investigate addresses that exhibit each type of RTT and find potential explanations.

3.5.1 Are satellites involved?

A reasonable hypothesis is that satellite links, widely known for their necessarily high minimum latency, would also be responsible for very high maximum latencies. Transmissions via geosynchronous satellite must transit 35,786km to a satellite and back, leading to about 125 ms of one way delay [61, 62]. Another 125 ms for the return trip yields a theoretical minimum of 250ms.

We expect satellite ISPs to have high 1st percentile latencies, but we consider whether they have high 99th percentile latencies as well. We use data from ISI survey IT63c (20150206) for this analysis because it provides hundreds of ping samples per IP address, and we wish to study relatively few addresses in some detail. Figure 3.11 shows the plot of 1st percentile latencies vs. the 99th percentile latencies for addresses in this survey. We separate addresses that the Maxmind database maps to known satellite providers, including Hughes and ViaSat. At left, we show the overall distribution without addresses from known satellite ISPs; at right, we show only satellite ISPs. (Recall that the precision of values just above the ISI timeout of three seconds is limited to integers; this creates the horizontal bands.) The satellite-only ISPs plot shows that the 1st percentile RTT for these addresses exceeds 500ms in all cases, showing that the RTTs are almost double the theoretical minimum. There are some points in the left plot that remain within the satellite-like cluster; at least some of these are from rural broadband providers that provide both satellite and other connectivity, such as xplorenet in Canada, which had at least one IP address report with a below 0.5s first percentile.

Each satellite provider has a distinct cluster in this scatter plot, and two smaller providers, Horizon and iiNet, have clusters of reports that produce near-horizontal lines in the graph, with varying 1st percentile but fairly consistent 99th

		May 2015			June 2015			July 2015		
ASN	Owner	>1s	%	Rank	>1s	%	Rank	>1s	%	Rank
26599	TELEFONICA BRASIL	3.56M	80.4	1	3.87M	77.5	1	4.20M	77.0	1
26615	Tim Celular S.A.	1.35M	74.5	3	1.42M	71.5	2	1.72M	71.6	2
45609	Bharti Airtel Ltd.	1.46M	76.6	2	1.21M	81.0	3	1.03M	79.2	3
22394	Cellco Partnership	0.55M	73.4	8	0.58M	73.5	4	0.63M	72.7	4
1257	TELE2	0.67M	69.5	5	0.42M	65.5	9	0.58M	67.4	5
27831	Colombia Movil	0.53M	68.8	9	0.54M	64.3	5	0.53M	62.8	6
6306	VENEZOLAN	0.69M	77.3	4	0.41M	76.4	10	0.40M	75.7	10
9829	National Internet Backbone	0.57M	27.6	7	0.43M	30.9	7	0.43M	29.5	9
4134	Chinanet	0.60M	1.5	6	0.38M	0.9	11	0.34M	0.9	11
35819	Etihad Etisalat (Mobily)	0.42M	54.0	10	0.43M	54.5	6	0.45M	55.8	8

Table 3.4: Autonomous Systems sorted by the addresses summed across three Zmap scans for addresses that observed RTTs greater than 1s. The table shows for each AS: the number and percentage of addresses with RTT greater than 1s and the rank in that scan.

percentile, as if queuing for these addresses is capped but the base distance to the satellite varies by geography.

Although some satellite hosts do have remarkably high RTT values—up to 517s—their 99th percentile values are predominantly below 3s. They do not have such high 99th percentile values as the rest of the hosts with over 0.3s first percentiles (those shown on the left graph). Thus, satellite ASes accounted for very few of the high latency addresses.

3.5.2 Autonomous Systems with the most high latency addresses

Next, we investigate the ASes and geographic locations with the most high latency addresses to identify relationships. For this analysis, we use Zmap scans from 2015 to identify high latency addresses. Zmap pings every IPv4 address,

	May 2015		**June 2015**		**July 2015**	
Continent	**>1s**	%	**>1s**	%	**>1s**	%
South America	7.27M	26.7	7.41M	25.8	8.05M	26.9
Asia	5.56M	3.8	4.73M	3.4	4.56M	3.2
Europe	2.56M	2.7	2.09M	2.2	2.32M	2.4
Africa	1.12M	29.4	1.20M	30.3	1.30M	31.7
North America	0.93M	1.0	1.04M	1.1	1.14M	1.2
Oceania	0.08M	3.9	0.08M	3.7	0.08M	3.7

Table 3.5: Continents sorted by the addresses summed across three Zmap scans for addresses that observed RTTs greater than 1s. The table shows for each AS: the number and percentage of addresses with RTT greater than 1s in that scan.

thereby covering addresses from all ASes. We chose the May 22, Jun 21 and Jul 9 Zmap scans to study. These scans were conducted at different times of the day, on different days of the week and in different months, as shown in Table 3.3. For each of these Zmap scans, we use Maxmind to find the ASN and geographic location for every address that responded.

ASes most prone to RTTs greater than 1 second

Figure 3.7 showed that the percentage of addresses that sent high latency Echo Responses remained stable over time. In particular, around 5% of addresses observed RTTs greater than a second in each scan. We refer to these addresses as

		May 2015			June 2015			July 2015		
ASN	Owner	>100s	%	Rank	>100s	%	Rank	>100s	%	Rank
26599	TELEFONICA BRASIL	51.9K	1.2	1	63.5K	1.3	1	77.6K	1.4	1
12430	VODAFONE ESPANA S.A.U.	12.8K	4.4	2	11.6K	4.1	2	14.6K	5.2	3
26615	Tim Celular S.A.	6.2K	0.3	7	9.4K	0.5	3	14.7K	0.6	2
3352	TELEFONICA DE ESPANA	8.5K	0.2	3	7.3K	0.1	5	7.5K	0.2	4
6306	VENEZOLAN	7.5K	0.8	5	8.4K	1.5	4	6.6K	1.2	6
22394	Cellco Partnership	6.9K	0.9	6	6.6K	0.8	6	7.5K	0.9	5
27831	Colombia Movil	3.2K	0.4	10	5.0K	0.6	7	5.2K	0.6	7
45609	Bharti Airtel Ltd.	7.8K	0.4	4	2.6K	0.2	9	2.9K	0.2	9
35819	Etihad Etisalat (Mobily)	3.8K	0.5	9	3.9K	0.5	8	4.0K	0.5	8
1257	TELE2	6.2K	0.4	8	1.7K	0.3	14	2.4K	0.3	12

Table 3.6: Autonomous Systems sorted by the addresses summed across three Zmap scans for addresses that observed RTTs greater than 100s. The table shows for each AS: the number and percentage of addresses with RTT greater than 100s and the rank in that scan.

turtles and investigate their distribution across Autonomous Systems to identify relationships.

For each Zmap scan, we found the turtles and identified their AS, and ranked ASes by the number of contributed turtles. Finally, we summed the turtles from each AS across the three scans and sort ASes accordingly and show the top ten in Table 3.4. For example, AS26615 had the second-largest sum of turtles across the three Zmap scans, but was ranked third within the May 2015 scan.

Inspecting the owners of each of these Autonomous Systems reveals that a majority of them are cellular. AS26599 (TELEFONICA BRASIL), a cellular AS in Brazil, has the most turtles, more than double that of the next largest AS in each of the scans. The next two ASes, AS45609 (Bharti Airtel Ltd.), and AS26615 (Tim Celular), are also cellular, and so are 5 of the remaining 7 ASes in the top 10.

Also notable is the percentage of responding addresses that are turtles for these ASes. Most of the cellular ASes have around 70% of all probed addresses being turtles. AS9829, one of the two ASes with turtles accounting for lower than 50% of probed addresses, is known to offer other services in addition to cellular. AS4134, with only 1% of its probed addresses being turtles, is also known to offer other services. We believe that the cellular addresses observe high RTTs while others do not, explaining the low ratio of probed addresses with RTTs greater than 1 second.

Finally, nine ASes were observed in the top ten in every scan. AS4134 was the only exception, but it ranked 11th in the June and July scans. Thus, the Autonomous Sytems with the most turtles also remain consistent over time.

Table 3.5 shows the continents with the most turtles. South America and Asia alone account for around 75% of all turtles. Further, around a quarter of all addresses in South America and a third of the addresses in Africa experienced RTTs greater than 1s in each scan. On the other hand, only 1% of North America's addresses are turtles (of which more than half come from a single ASN: AS22394).

ASes most prone to RTTs greater than 100 seconds

Next, we investigate the Autonomous Systems of addresses with RTTs greater than 100 seconds in the three Zmap scans: we refer to these addresses as sleepy-turtles. We consider whether these addresses are different from turtles to identify whether there is a different underlying cause. Following the same process in identifying ASes and sorting them as in Table 3.4, Table 3.6 shows Autonomous Systems that are most prone to RTTs greater than 100 seconds.

We find that sleepy-turtles exhibit similarities to turtles. Every Autonomous System in Table 3.6 is cellular. Further, the ranks of the Autonomous Systems remain stable over time across the scans. However, there is more variation across the scans for the percentage of sleepy-turtles among all probed addresses for an AS. This suggests that the fraction of addresses experiencing RTTs greater than 100 seconds is less stable over time.

3.5.3 Is it the first ping?

RTTs that are consistently greater than a second are sufficiently high that interactive application traffic would seem impractical with these delays. We suspected that the latencies measured by ISI and Zmap might not be typical of application traffic.

We considered two broad explanations—extraordinary persistent latency due to oversized queues associated with low-bandwidth links, or extraordinary temporary, initial latency due to MAC-layer time slot negotiation or device wake-up.

In this section, we find that the latter appears to be a more likely explanation, qualitatively consistent with prior investigations of GPRS performance characteristics [63], but showing quantitatively more significant delay.

We extracted 236,937 IP addresses from the 20150206 ISI dataset (February 2015), including all addresses with a median RTT of at least one second. To select only responsive addresses that still had high latency, for each of these IP addresses, we sent two pings, separated by five seconds, with a timeout of 60 seconds. We omit 151,769 addresses that did not respond to either probe and 1,994 addresses that responded, on average, within 200ms.

Of the 83,174 addresses that remain, we wait approximately 80 seconds before sending ten pings, once per second with the same 60-second timeout. We next classify how the round trip time of the first ping, RTT_1, differs from those of the rest of the responded pings, $RTT_2 \dots RTT_n$, where n may be smaller than 10 if responses are missing. For most of these addresses, 51,646, the first response took longer than the *maximum* of the rest. This suggests that roughly 2/3 of high latency observations are a result of negotiation or wake-up rather than random latency variation or persistent congestion. For 11,874, $median(RTT_2 \dots RTT_n) < RTT_1 < max(RTT_2 \dots RTT_n)$, i.e., the first response took longer than the *median*, but not the maximum, of the rest. The first response was smaller than the median of the rest for a comparable 10,910. That the first is above or below the median in roughly equal measure suggests that for these addresses there is little observed penalty to the first ping. Finally, we omit analysis of 8,329 addresses because we did not receive a response to, at least, the first probe, even though they did

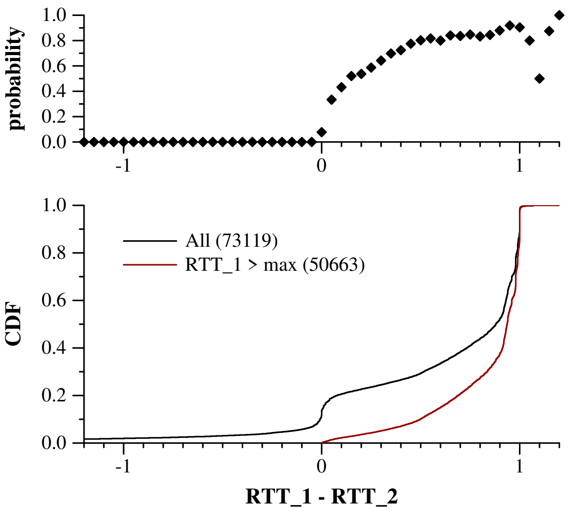


Figure 3.12: Bottom: Difference between initial latency and second probe latency; values around 1 indicate that both responses arrive at about the same time, values near zero indicate that the RTTs were about the same. The second line includes only those where $RTT_1 > max(RTT_2 \ldots RTT_n)$. Top: The probability that, given $RTT_1 - RTT_2$ on the x-axis, that $RTT_1 > max(RTT_2 \ldots RTT_n)$.

respond to the initial pair of probes, and we omit an additional 415 addresses that did respond to the first probe, but not to at least four probes overall (i.e., we require $n \geq 4$ before computing the median or maximum for comparison).

Can the overestimate be detected?

We show in Figure 3.12 the differences between the first and second round trip times for all those that had a first and second response. (1,311 addresses responded to the first but not the second). Rarely, latency increases from first to the second (yielding a negative difference) or decreases sufficient to indicate reordering (yielding a difference greater than one second). Typical among these

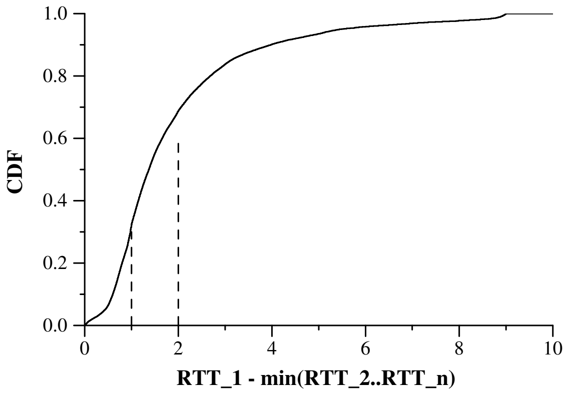


Figure 3.13: Difference between initial latency and observed minimum. The typical setup time is below four seconds.

addresses is for the second ping to be one second less than the first, that is, for both responses to arrive at about the same time.

We infer that a measurement approach that sent a second probe after one second could detect this behavior. The top graph of Figure 3.12 shows the probability that the maximum will be less than the first based on the difference between the first two latencies. (When the RTT difference exceeds 1 at the right edge of the upper graph, there are very few samples in an environment of substantial reordering.) Any significant drop from RTT_1 to RTT_2 is indicative of an overestimate with high probability.

How long does the negotiation or wake-up process take, and how large is the overestimate?

We observe that this can be estimated by comparing the first round trip time to the lowest seen among the ten probes. Of course, if the negotiation takes 15 seconds,

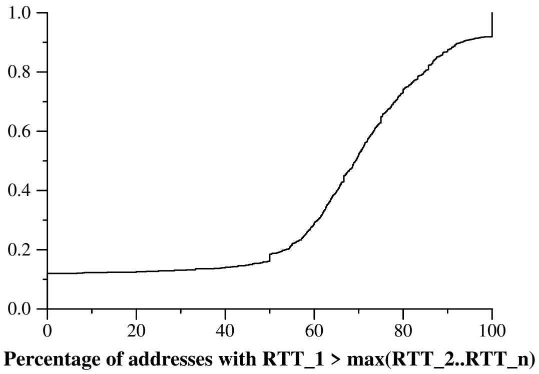


Figure 3.14: Percentage of addresses in a /24 prefix showing a drop from the initial to the maximum.

the first probe rtt will take at most 9 seconds longer than the last, so this data set will treat all instances of a setup time between 10 and 60 seconds as taking 9. We show in Figure 3.13 the differences between RTT_1 and $min(RTT_2 \ldots RTT_n)$ for those 51,646 addresses that had a higher first rtt than the maximum of the rest. The median is 1.37 seconds, and 90% of the differences are below 4 seconds. Only 2% of the samples are above 8.5 seconds, suggesting that we do not underestimate this time substantially, and thus conclude that the wake-up or negotiation process generally takes from one-half to four seconds.

Are the addresses that show a high initial ping scattered across the IP address space or clustered into /24s?

The 236,937 IP addresses that we decided to probe initially are from only 1,887 "/24" prefixes. This is somewhat fewer prefixes than would be expected, given that there are 3.6M addresses in 34K prefixes in the overall 20150206 dataset. That

Pattern	Pings	Events	Addrs
Low latency, then decay	615	13	10
Loss, then decay	1528	81	33
Sustained high latency and loss	2994	21	14
High latency between loss	12	12	12

Table 3.7: We observed distinct patterns of latency and loss near high latency responses, classifying all 5149 pings above 100 seconds from the sample.

is, as one might expect, greater than one second latencies do seem to be a property of the networks associated with selected prefixes. The 83,174 addresses that responded are from only 1,230 prefixes. We show the percentage of responsive addresses within each prefix that dropped from the initial ping to the maximum of the rest in Figure 3.14. Several prefixes did not have an initial latency greater than the maximum; these typically had very few responsive addresses. In other prefixes, most addresses showed a reduction. Finally, the 51,646 that showed a reduction from the initial ping are from only 1,083 prefixes. Of the 161 prefixes that had only one address with above one-second median latency, only 39 showed a reduced from the initial RTT to the maximum of the rest. Taken together, we believe this distribution of addresses across relatively few prefixes indicates that the wake-up behavior is associated with some providers but not restricted to them.

3.5.4 Patterns associated with RTTs greater than 100 seconds

Finally, we look at addresses with extraordinarily high latencies (greater than 100 seconds); in particular, we want to understand whether these high latencies are an instance of a first-ping-like behavior, where wireless negotiation or buffering during intermittent connectivity creates the high value, or, on the other hand, are instances of extreme congestion. To separate the two types of events, we consider a sequence of probes, looking for whether or not the latency diminishes after a ping beyond 100 seconds.

We sample 3,000 of 38,794 addresses whose 99th percentile latency was greater than 100 seconds in the IT63c (20150206) dataset. Of this sample, 1,400 responded. We sent each address 2000 ICMP Echo Request packets using Scamper, spaced by 1 second. To collect responses with very high delays without altering the Scamper timeout, we simultaneously run tcpdump to capture packets.

Ping samples that saw a round trip time above 100 seconds exist in the context of a few very distinct patterns. Often, a series of successive ping responses would be delivered together almost simultaeously, leading to a steady decay in their round trip times. For example, after 136 seconds of no response from IP address 191.225.110.96, we received all 136 responses over a one second interval: every subsequent response's round-trip latency was 1 second lower than the previous. This pattern is sometimes preceded by a relatively low latency ping ($<$ 10 seconds) and at other times, follows a few lost pings: we distinguish between these two cases and call the former *Low latency, then decay* and the latter *Loss, then* *decay*. It is possible that these are both observing the same underlying action on the network, but we leave them separate since there are substantially many of each.

Another characteristic pattern is that a high round trip time is followed by several responses of even greater latency, possibly with intermittent losses. This behavior is usually sustained for several minutes with latencies remaining higher than normal (>10 seconds) throughout the duration: we call this behavior *Sustained high latency and loss*. Finally, there are some cases where a single ping has a latency $>$ 100 seconds and is preceded and followed by loss. We call these cases *High latency between loss*.

We count the number of occurrences of each pattern in Table 3.7. For each pattern, we show the number of pings greater than 100 seconds that were part of that pattern, the number of instances of that pattern occurring, and the number of unique addresses for which it occurred. We observe that the majority of events and addresses are *Loss, then decay*, yet almost twice as many pings are part of *Sustained high latency and loss*.

3.5.5 Summary

High latencies appear to be a property mainly of cellular Autonomous Systems, though a few also appear on satellite links. Latencies in the ISI data that are regularly above one second seem to be caused by the first-ping behavior associated with several addresses, where the first ping in a stream of pings has higher latency than the rest. Egregiously high latencies, i.e., latencies greater than a hundred seconds, occur in two broad patterns. In the first, latencies steadily decay with each probe, as if clearing a backlog. In the second, latencies are continuously high and are accompanied by loss, as if the network link is oversubscribed.

3.6 Conclusion and Discussion

Researchers use tools like ping to detect network outages, but generally guessed at the timeout after which a ping should be declared "failed" and an outage suspected. The choice of timeout can affect the accuracy and timeliness of outage detection: if too small, the outage detector may falsely assert an outage in the presence of congestion; if too large, the outage detector may not pass the outage along quickly for confirmation or diagnosis.

We investigated the latencies of responses in the ISI survey dataset to determine a good timeout, considering the distributions of latencies on a per-destination basis. Foremost, latencies are higher than we expected, based on conventional wisdom, and appear to have been increasing. We show that these high latencies are not an artifact of measurement choices such as using ICMP or the particular vantage points or probing schemes used, although different data sets vary somewhat. We show that high latencies are not caused by links with a substantial base timeout, such as satellite links. Finally, we showed that in many instances, the initial communication to cellular wireless devices is largely to blame for high latency measures. Similar spikes that may be consistent with handoff also dissipate over time, to more conventional latencies that support application traffic. With this data, researchers should be able to reason about what to expect in terms of false outage detection for a given timeout and how to design probing methods to account for these behaviors.

As memory capacity and performance becomes less of a limiting factor, we believe that the lesson of this work is to design network measurement software to approach outage detection using a method comparable to that of TCP: send another probe after 3 seconds, but continue listening for a response to earlier probes, at least for a duration based, at least in part, on the error rates implied by Table 3.2.

When investigating historical outage measurement data collected by probing based techniques, the timeouts used by the technique must be compared with timeouts that would have captured almost all responses for the addresses pinged by the technique. For example, Thunderping probes addresses only in U.S. networks. For these addresses, both the ISI and Zmap datasets showed that more than 99% of ping responses arrived within the 5s timeout used by Thunderping. The probability of false outage inferences due to high latency is small. However, if the Thunderping technique had been used to ping addresses in South American cellular ISPs, there would be a significantly higher probability of detecting false outages, since the 5s timeout would have missed many delayed responses.

Chapter 4: Mitigating false inferences due to dynamic addressing

In this chapter, I begin by describing a common assumption—that IP addresses can be used as proxies—for users in Section 4.1. In Section 4.2, I discuss how dynamic addressing can lead probing-based outage detection techniques to make false inferences about outages.

Next, I describe work with colleagues to empirically measure the frequency of dynamic addressing and the durations for which addresses are assigned to residential home router devices in several networks around the world and the effect of outages upon dynamic reassignment [21]. The measurements we used are sourced from the RIPE NCC's Atlas project, which deploys small devices, called probes, that conduct measurements from globally distributed networks [15]. The RIPE Atlas dataset offers measurements that allow us to determine when an IP address change occurred and what the addresses were before and after the change. In addition, the dataset includes many measurements that provide context about what was happening around the time of the address change. I was able to use these measurements to detect when RIPE Atlas probes rebooted and were not sending pings (indicating a power outage) and when their pings were not getting responses (indicating a network outage). In a study with colleagues of active RIPE Atlas probes in 2015, we found 3,038 RIPE Atlas probes with address changes hosted across 929 ISPs and 156 countries [21]. Using the measurements from RIPE Atlas, I identify networks where addresses are typically stable.

Finally, in section 4.9 I discuss a technique to identify outages even in networks where dynamic reassignment is common. Using a complementary dataset that allows checking if an address for which a probing-based technique detected an outage has remained the same before and after the detected outage, we are able to confirm outages even in networks where dynamic reassignment is common.

4.1 IP addresses can be proxies for end users

Academia and industry often rely on a simplifying assumption that IP addresses uniquely identify end-hosts [64–78]. This assumption allows researchers to track end host behavior over time [5, 68, 69], or to count participating users in peer-to-peer systems [64–66]. Many organizations create blacklists of suspicious IP addresses based on previously observed malicious traffic associated with those addresses [75–78].

Probing-based techniques like Thunderping [5] make a similar assumption: a probed address is representative of a residential customer's Internet connection. Many residences have at least one device with a public IP address [20], typically the home router. When a home router's address stops responding to pings, it could be evidence of a residential Internet outage.

All of these applications would benefit from understanding how often and when dynamic addresses assigned to user devices change.

4.2 Probing-based techniques can make false outage inferences due to dynamic addressing

When probing-based remote outage detection techniques send probes to an address, they expect that the address continues to be assigned to the same end-host for the entirety of the probing duration. Depending upon how a dynamic address gets reassigned, these techniques can make false inferences about outages in two ways:

- *Detecting false outages* Probing-based remote outage detection techniques detect outages when a previously responsive address stops responding to probes. However, If a dynamic address being probed is withdrawn from its host and is not assigned to any other host, active probes to the address will no longer elicit responses. These techniques will infer false probe-loss, leading them to infer false outages.

- *Detecting false outage duration* These techniques detect outage duration by continuing to probe an unresponsive address. When the address starts responding to probes again, the outage is inferred to end. If a home router with a public dynamic address has an outage and at some point during the outage, the dynamic address is reassigned to some other home router which responds to probes, probing-based remote outage detection techniques would infer that the outage ended incorrectly.

My approach to mitigating these false inferences is to analyze how frequently and for what reasons dynamic addresses are reassigned, in various networks. Using the results of these analysis, I identify networks where addresses are typically stable.

4.3 Dynamic addressing background

An IP address can be used to uniquely identify the end-host it is assigned to until the end-host's address changes for some reason. The duration of time that a dynamic IP address continues to be assigned to the same CPE (Customer Premises Equipment) device depends upon various causes that can induce the assigned IP address to change. Here, I present techniques used for assigning dynamic addresses and the events and agents involved in dynamic address changes.

4.3.1 Dynamic Host Configuration Protocol

ISPs often use the Dynamic Host Configuration Protocol (DHCP) [79] for IP address assignment. DHCP issues an IP address to a host for a lease duration configured by the ISP. The host will try to renew the lease before it expires, typically half-way into the lease. However, whether the same IP address is renewed, or a different one is assigned, depends upon ISP policy. We speculate that the typical behavior of ISPs using DHCP is to renew the lease of the currently assigned IP address, since one of the stated design goals in the DHCP specification is that a DHCP client should be assigned the same address in response to each request, whenever possible. Thus, we typically only expect an ISP using DHCP, to change the address of a CPE, if something happens to prevent the CPE from renewing its lease (like an outage).

4.3.2 Point-to-Point Protocol

In some networks, end-hosts connect to an ISP using point-to-point links. For these networks, the Point-to-Point Protocol (PPP) first configures and establishes the point-to-point link [80]. Next, a Network Control Protocol (NCP) like the Internet Protocol Control Protocol (IPCP) configures IP addresses [81]. The PPP specification notes that the link will remain configured for communication until the link is actively closed down through network administrator intervention or when an inactivity timer expires.

4.3.3 Potential dynamic address change causes

Next, we identify the reasons dynamic addresses assigned using the above techniques could change. We classify the following categories of address change:

- **Changes after outages** If the client is disconnected or loses power long enough to fail to renew a DHCP lease, its address may be assigned to another; when it returns, it may then get a new address. We call such changes *outage-caused address changes*.

- **Changes after reboot/reconnect** While we expect addresses assigned through traditional DHCP to change only when the outage duration is long enough to prevent lease renewal, addresses assigned through PPP can change upon outages of any duration. Any reboot or network reconnect event could cause the client to forget its prior address and request a new one, or the state associated with a connection may be lost. We call such address changes *reboot-caused address changes*.

- **Administrative address changes** A purpose of dynamic address assignment is to allow reconfiguration of the network; it is possible that a reconfiguration of the DHCP server will force a change to the subnet on which the client lies. We expect such reassignment to be rare.

- **Periodic address changes** We observe that some ISPs limit the session length associated with an address, causing a reassignment after a fixed duration, typically one day to one week depending on the ISP.

Intuitively, the address change is either caused by the ISP (administrative or periodic), or caused by the client (or an interruption in network service to the client) in a reboot or outage.

4.4 Related Work

Previous work studied the performance of DHCP in small campus networks [82, 83] and settings where smartphone usage is widespread [84] and developed techniques to reduce network address utilization and DHCP broadcast traffic. The goal of those studies was to improve the performance of DHCP by tuning configuration.

Conceptually, so long as there is some uniquely identifying feature that remains constant across a host's address change, it is possible to track IP address changes over time for that host. Several studies have used this broad method [44, 47, 83, 85–89]. UDmap [85] studied dynamic address properties using Hotmail user login traces where the user's login serves as the identifying feature. Casado et al. [88] tracked clients using HTTP cookies when clients access a CDN. Other studies [47, 87] used continuous responsiveness of an address itself as the identifying feature, assuming that an address that responds continuously belongs to the same user and that when an address stops responding to pings, it has been reassigned.

While we share the same goal as these studies, our approach diverges in that we are interested in the events associated with an address change. Previous studies lacked access to end-host information that could reveal the cause of an address change. One exception, Maier et al. [87], used access to the Radius server of a European DSL provider from one urban area to identify why DSL sessions terminated, and noted that the DSL provider often limited Radius session length to 24 hours in that area. We extend this result to several ISPs in countries from Europe, Asia, and South America, and identify other typical session length limits. Argon et al. [44] used periodic measurements from end-hosts in the DIMES infrastructure [18]. DIMES software installed on an end-user computer is different from RIPE Atlas hardware probes primarily in that it reports back only every 30-60 minutes (as opposed to RIPE Atlas's 3 minutes), the agent can be installed on laptops that move (as opposed to RIPE Atlas probes that could move, but do not), the hosts running DIMES are often powered down (resulting in limited uptime), and DIMES hosts appear to have static IP addresses more often (they reported 60% had only one address). Nevertheless, Argon et al. observed that some small ISPs exhibited address alternation with a 24 hour periodicity. In IPv6, the RFC for privacy extensions for stateless address autoconfiguration recommends that IPv6 addresses be changed every 24 hours [90] and empirical results by Plonka and Berger found that more than 90% of client IPv6 addresses were ephemeral [91]. We showed that 24 hour defaults are not uncommon in IPv4 as well.

These studies relied on relatively uncontrolled observations of the address assigned to a device or user, both in terms of whether the devices are active, whether the users connect using multiple devices, and how frequently samples are provided. As a consequence, the dynamic IP address churn rates reported by these studies vary. While UDmap reported that over 30% of IP addresses have inter-user durations of 1–3 days [85], Heidemann et al. reported that 90% of IP addresses were occupied for less than a day [47]. Maier et al. [87] reported that a major European ISP had per-user median durations of just 20 minutes during their study in 2009 (we did not observe this duration in 2015). We believe that the perspective of a device using the dynamically assigned network is necessary for understanding the reasons behind the address change and for getting precise information about the duration that any address is held. Further, since RIPE Atlas probes provide continuous, longitudinal measurements enabling the inference of successive addresses assigned to a CPE device, we perform the first analysis of dynamic prefixes from which devices are assigned successive addresses.

4.5 The RIPE Atlas datasets

Analyzing periodic and administrative address changes requires visibility of the dynamic addresses assigned to a sample of the ISP's customers and the ability to see these addresses change over time. Analyzing outage-caused and reboot-caused address changes requires knowledge of the events occurring on the end-host at the time of an address change. Prior studies of dynamic addressing have typically relied on incoming connections that have a unique client identifier, such as a user name, but changing addresses, and thus have no information about what caused a change or precisely when it occurred. The RIPE Atlas dataset is unique since it includes necessary information about both address changes and contemporaneous events at the host.

The RIPE NCC's Atlas project deploys small devices, called probes, that conduct measurements from globally distributed networks [15]. In this section, we first describe the connection logs dataset from RIPE Atlas that we use to detect

IP address changes. We then describe the k-root ping and SOS-uptime datasets from RIPE Atlas that we use to learn about events occurring on end-hosts.

4.5.1 RIPE Atlas connection logs dataset

RIPE Atlas probes connect to the RIPE Atlas infrastructure through a single SSH session over TCP port 443 (typically used by HTTPS) [92]. RIPE Atlas servers record the establishment and termination of these connections in *connection logs*. Table 4.1 shows connection log entries for a RIPE Atlas probe in the dataset for the first five days in January 2015.

Connection logs record each TCP connection made by the probe to a central controller and include the timestamp of the beginning and end of the connection (defined by the last receipt of data), the peer address of the connection that represents the publicly visible IP address used by the probe, and a unique identifier of the probe device. Probes are typically deployed behind the Customer Premise Equipment (CPE) of a user, so that the publicly visible IP address appearing in the connection logs belongs to that of the CPE. We term this address the “probe’s address” or the “end-host address,” since it is the useful, publicly visible address that the probe uses, even though the address may technically belong to the CPE and the probe has a different, private, RFC 1918 address.

We find IP address changes by inspecting these connection logs. A new entry in a probe’s connection log is created whenever an event occurs that causes the existing TCP connection to break. This connection will break when the probe’s

ID	Start time	End time	IP Address	Dur
206	Dec 31 03:21:34	Jan 1 02:57:37	91.55.174.103	NA
206	Jan 1 03:22:16	Jan 1 17:34:11	91.55.169.37	14.2
206	Jan 1 18:00:54	Jan 1 18:42:31	91.55.132.252	0.7
206	Jan 1 19:06:46	Jan 2 02:19:16	91.55.155.115	7.2
206	Jan 2 02:41:55	Jan 3 02:18:00	91.55.141.95	23.6
206	Jan 3 02:43:14	Jan 4 02:16:59	91.55.165.167	23.6
206	Jan 4 02:40:58	Jan 5 02:15:45	91.55.163.252	23.6
206	Jan 5 02:38:39	Jan 6 02:14:48	91.55.141.63	NA

Table 4.1: Connection log sample for the first five days of 2015. We compute the address duration, shown in the last column in hours.

IP address changes, when a probe reboots, or when there is an outage. We can infer that the address changed between the end time of one connection and the start time of the next, if the addresses differ in consecutive entries. For example, in Table 4.1, there are seven address changes. Between changes, we can identify the duration that the probe held an address, shown in hours. In this example, each connection had a different address, so the address durations are equal to the connection duration, though this is not always the case. The duration of the first address is unknown because we do not know when that IP address was first assigned to the probe; the duration of the last address is also unknown.

The interval between connections, in the example of Table 4.1, typically 20–25 minutes, is information we also use in concert with other datasets described below to determine the type and duration of the event that led to a new connection. An active RIPE Atlas probe should report experiments back to the central controller about every three minutes [93]. We attribute this long delay between the end of one connection to the beginning of the next when there is an address change to waiting for TCP to exhaust its retransmission attempts (RFC 1122 Section 4.2.3.5) [49].

We obtained connection logs from January 1, 2015 to December 31, 2015 belonging to 10,977 active RIPE Atlas probes that had been connected to their central controllers for more than 30 days in 2015. We first found the list of active probes as of December 31, 2015, using the RIPE Atlas probe archive [94], and found 16,584 active probes. Next, we scraped each active probe's connection logs directly from the probe's webpage [95]. Subsequently, we found 10,977 probes who had been connected to their central controllers for an aggregate duration of more than 30 days in 2015.

4.5.2 Probe filtering

We omit from our analysis two sets of data: probes that are connected using a method where using different addresses does not indicate changes to the addresses that were assigned, for example, multihomed probes, as well as connection log entries that represent movement from one location or provider to another. Once we omit a probe for anomalous behavior in connection logs, we omit that probe from our analysis of the other RIPE Atlas datasets as well.

Table 4.2 provides an overview of the probes we omitted from the analysis.

IPv6 and dual-stacked probes

Probes that communicate, even occasionally, over IPv6 are not useful for under-

Category	**Probes**
Total Probes	10,977
Not Analyzable	
Never changed	3,073
Dual Stack	3,728
IPv6	237
Multihomed / Core / Data-center (tags)	174
Multihomed (alternating addresses)	511
Only address change from 193.0.0.78	216
Analyzable (geography)	**3,038**
Multiple ASes	766
Analyzable (AS-level)	**2,272**

Table 4.2: Of the 10,977 probes in the dataset, we are able to find address changes on 3,038 probes. 766 probes had addresses from multiple ASes; we discard address changes across ASes for these probes from our geographic analysis and filter these probes altogether in our AS-level analysis.

standing IPv4 address dynamics. We found 237 probes that made connections solely over IPv6 and 3,728 that used both IPv4 and IPv6. The 3,728 that connect over both protocols often alternated between address types, providing little information about the duration that the probe held any particular IPv4 address. Concretely, if a dual-stacked probe established one TCP connection to the central controller over IPv4 and the next TCP connection over IPv6, we cannot tell whether or when the IPv4 address changed while the IPv6 connection was active. We would need consecutive IPv4 connections from three different IPv4 addresses to determine how long the probe held the address in the middle of the sequence. In practice, a sequence of such IPv4 connections is rare for a dual-stack probe.

Multihomed and datacenter probes

We cannot use the connection logs dataset to observe address changes accurately on multihomed probes (probes that have more than one available IP address concurrently). For these probes, a connection from a new address could simply be a connection from the other address assigned to the CPE, much like a dual-stack probe. Probes at exchange points or in data centers are relatively few and seemed more likely to be problematic (by exhibiting multihomed behavior) than instructive (by representing address changes experienced by customers).

To filter multihomed probes, we first looked for hints in user-provided "tags" associated with a probe: 174 probes had at least one of the tags "multihomed," "datacentre," or "core." Tags are provided voluntarily and so probes may not be tagged with those labels even if they were in fact multihomed; thus, we looked for common features among the tagged probes which we could then use to omit probes with similar behavior. The most common feature we found was that connections from the tagged probes alternated between one fixed address and another potentially changing address; we found this feature on 36 of the 174 tagged probes. We found 511 other probes that matched this behavior and removed them from the dataset. We expect that it is far more likely that when a host returns to using a previously-used address, the host is choosing from among addresses it holds for a long time rather than that the ISP reassigned a previously held address to the host. We combine this behavioral, alternating-addresses, definition of multihomed with the tags to choose probes to omit from analysis.

4.5.3 Connection log entry filtering

We omit some entries in the connection log because of properties of either the address involved or because the detected address change was such that a probe reported an address from one autonomous system for one connection and an address from a different autonomous system for the next connection. Removing these connection log entries does not generally remove probes entirely from analysis.

Testing addresses

Some probes had their first address transition from the same IP address, 193.0.0.78. This address belongs to the RIPE NCC, and was used for testing before being shipped to volunteers. There were 427 such probes that started with this address; we remove this connection log entry. That left 216 additional probes with no further address changes in 2015, so we omitted those probes in Table 4.2.

Address changes across ASes

When attributing behavior to individual autonomous systems, we omit from analysis any probes where address changes indicated a change from the address space of one autonomous system to the address space of another. We used CAIDA's IP-to-AS dataset [96] to map each IP address to its autonomous system. CAIDA publishes the IP-to-AS dataset monthly; thus, we found the month in which a new IP address was assigned to a probe and used CAIDA's IP-to-AS dataset for that month to find the AS for that address. We found 766 probes with at least one address change spanning different autonomous systems. These ASes could be sibling ASes owned by the same ISP, but could also belong to different ISPs if the owner of the probes switched ISPs. For our geographic analysis (Section 4.6.2), we discarded the address changes spanning ASes for these probes, but retained the address changes within the same AS. For our AS-level analysis of renumbering behavior (Section 4.6.3), we made the conservative choice of filtering these probes altogether.

Table 4.2 summarizes the dataset and the number of probes filtered. After the filtering process we had 2,272 probes analyzable for AS-level renumbering behavior, and 3,038 probes analyzable for geographic renumbering behavior. For each analyzable probe in Table 4.2, we found address changes along with the time of the address change and used them to find the duration for which addresses were assigned before changing.

4.5.4 k-root ping dataset

We detect network outages using two items from the built-in RIPE Atlas probe measurements. Every four minutes, each probe sends three pings to the k-root DNS server and logs the number of sent pings and the number of successful responses [97]. Table 4.3 shows a sample of this log. Probes report the results of

ID	Timestamp	N sent	N success	LTS
16893	Jan 27 09:01:42	3	3	86
16893	Jan 27 09:05:48	3	0	151
16893	Jan 27 09:09:45	3	0	388
16893	Jan 27 09:13:36	3	0	619
16893	Jan 27 09:17:49	3	0	872
16893	Jan 27 09:21:40	3	0	1103
16893	Jan 27 09:25:39	3	3	1342
16893	Jan 27 09:29:36	3	3	146

Table 4.3: Sample of k-root ping dataset for probe ID 16893 when a network outage occurred. We detect a network outage when pings to the k-root server are lost and when this ping loss is accompanied by increasing Last Time Synchronized (LTS) values. Here we detect a network outage beginning at Jan 27 09:05:48 and ending at Jan 27 09:21:40.

these and other measurements via HTTP POST to the central controller once every four minutes. Along with the measurement data, the probe also reports the current *LTS* or "last time synchronised" value. This value indicates when the probe last synchronized its clock with that of the central controller. Typically, probes synchronize their clocks by NTP or upon receipt of the HTTP verify response from the controller [93], so in the absence of an outage, the reported LTS value should be less than four minutes (240 seconds).

We use a combination of the ping responses and the LTS value to infer a network outage, so that we have two (mostly) independent measurements that indicate that the probe's network has failed. We consider the network outage to start at the first measurement where all pings to the k-root server were lost, and to end at the last measurement where all pings were lost. If the LTS value did not grow, that would indicate that the probe was still able to communicate with the controller, and thus would not be an outage. Note that this interval underestimates the duration of a network outage by up to eight minutes.

4.5.5 SOS-uptime dataset

ID	Timestamp	Uptime counter value
206	Jan 1 03:15:18	262531
206	Jan 1 17:50:26	315038
206	Jan 1 17:50:55	19
206	Jan 1 17:53:59	203
206	Jan 1 18:59:44	4147

Table 4.4: Sample of SOS-uptime records from RIPE Atlas for January 1 2015 for probe ID 206. The third row shows that the uptime counter had reset 19 seconds before 17:50:55, allowing us to infer that the probe rebooted at 17:50:36.

The SOS-uptime dataset contains probe uptime counter values over time. The uptime counter on each probe is 64 bits long and counts the number of seconds since the probe booted. Probes report their uptime counter value to the central controller every time they make a new TCP connection to the controller.

We use the SOS-uptime dataset to determine when RIPE Atlas probes rebooted by finding when the uptime counter was reset. For example, consider the sample SOS-uptime records from the RIPE Atlas dataset for probe ID 206 shown in Table 4.4. The first entry at 03:15:18 on January 1st shows that the probe had been up for 262,531 seconds. Later that evening, the probe is shown to have been up for 315,038 seconds, but the next uptime counter value reports that the probe was up for only 19 seconds. We infer that a reboot occurred 19 seconds earlier, at 17:50:36.

After finding reboot times, we use the k-root ping dataset to measure how long each power outage lasted. When we detect a reboot, we use the difference in time between successive pings to the k-root server to estimate the power outage duration.

4.5.6 Associating inter-connection gaps with outage events

The next task is to synthesize these three datasets to identify outage events that occur between TCP connections to the central controller. The TCP connection to the central controller breaks when the IP address changes, when the probe reboots, when the CPE reboots, or when there is a power outage or significant network outage. For example, the reboot at 17:50:36 in Table 4.1 corresponds to rows 2 and 3 in Table 4.1 since the reboot time falls between the end of the connection log entry ending at 17:34:11 and the start of the connection log entry beginning at 18:00:54.

We use a priority ordering to assign outages to inter-connection gaps. If the k-root dataset indicated a network outage in the gap, we associate it with a network outage. If instead the SOS-uptime dataset indicates a reboot coincident with missing attempted k-root pings from the k-root dataset, we associate the gap with a power outage. If neither occurred, we mark the gap as a "no-outage" indicating that the reconnection was not associated with any outage.

4.6 Periodic address changes

ISPs can assign dynamic addresses for as long as they wish. In DHCP, long leases simplify administration, while short leases can be more efficient in reclaiming unused addresses. DHCP leases, however, are meant to be renewable by devices that are still active. In this section, we look at periodic address reassignment: instances where a device changes address periodically, despite actively using the address. Periodic reassignment is atypical for devices using DHCP since a device that is continuously renewing its lease should continue to keep its current address [79].

4.6.1 Metric to detect periodic address durations

If ISPs intentionally renumber after specific durations, we would expect those address durations to be prominent in a distribution of all address durations belonging to that ISP. We initially considered studying distributions of raw address durations, similar to the analyses by Maier et al. [87] and Moura et al. [86], but found that short address-durations were overrepresented. For example, in Table 4.1, inspecting the cumulative distribution of address durations would suggest that only half the addresses (3 of 6) were assigned for 24 hours. However, when trying to reason about the expected duration that an address will continue to be assigned to the CPE, we would like to know the fraction of total time that each duration accounted for. For example, in Table 4.1, the CPE was assigned 24 hour long addresses for roughly three-quarters of the total measured time. This latter notion is more useful to find whether an ISP is using periodic durations consistently, since the modes at intervals on the scale of days will be more visible.

To capture this notion we define a metric, the *total time fraction*. For a given probe and an address duration d, we define the total time fraction for d as the fraction of time spent by the probe in durations of length d. We compute the total time fraction for a given probe and a duration d by obtaining the total address time for the probe, and computing the fraction of the total address time that was accounted for by address durations of length d. For a probe p, if $n(d)$ is the number of times the probe had an address duration d and D is an array containing all address durations that were assigned to the probe, the total time fraction for the address duration d is given by:

$f_d^p = d \times n(d)/\Sigma(D)$

We use a similar procedure for computing the total time fraction considering all probes in an ISP, country, or continent. We believe that the total time fraction offers a better representation of the probability that an address was assigned for a certain duration than a simple inspection of the address durations.

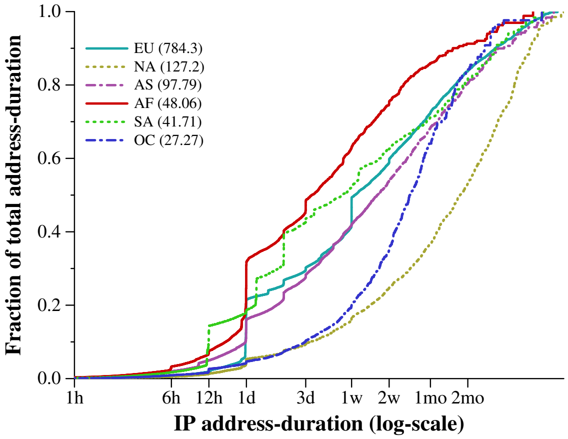


Figure 4.1: Cumulative distribution of total time fraction by continent. Modes (vertical segments in the CDF) indicate periodic renumbering. Addresses in North America are relatively long lived and free of periodic renumbering.

4.6.2 Periodic address changes by geography

We begin by inspecting how address durations vary across continents. We expected that address scarcity might affect address durations, leading to longer durations in North America and shorter durations in Asia. We use RIPE Atlas's probe database to find the country to which each probe belongs. Next, we aggregate the address durations of probes by their respective countries and subsequently, to their continents. Figure 4.1 shows the cumulative distribution of the total time fraction for each continent, i.e., the y-axis shows the fraction of total address duration accounted for by durations less than the x-axis value. The number in parentheses in the legend for each continent shows the total address duration for that continent in years ($\Sigma(D)$).

In Europe, Asia, Africa, and South America, address durations exhibit well-defined modes, mostly at intervals that are multiples of 24 hours. The most common mode is exactly at 24 hours: the total time fraction for European addresses at 24 hours is 0.16, African addresses is also 0.16, and Asian addresses is 0.07. One week address durations are also common in Europe, with the total time fraction at 1 week equaling 0.08. South American addresses exhibit multiple modes: their total time fraction is 0.11 at 12 hours, 0.07 at 28 hours, 0.09 at 48 hours, and 0.03 at 192 hours (8 days).

The curves for North America and Oceania do not have well-defined modes, suggesting that ISPs in these continents do not periodically change addresses. Further, North American probes typically retain their dynamic addresses for much longer durations than other continents; North American addresses spent more than half of the total time in address durations longer than 50 days. This suggests that IP addresses can be used as end-host identifiers in North America for several weeks.

4.6.3 Periodic address changes by AS

We next considered whether the configuration decision to renumber periodically was uniform across an AS, or could reflect some other feature. For example, periodic renumbering could be a result of an unexpected cron job on the RIPE Atlas probe or a faulty DHCP client that could not renew. Periodic renumbering could be due to government regulations in countries, perhaps as a privacy measure. It

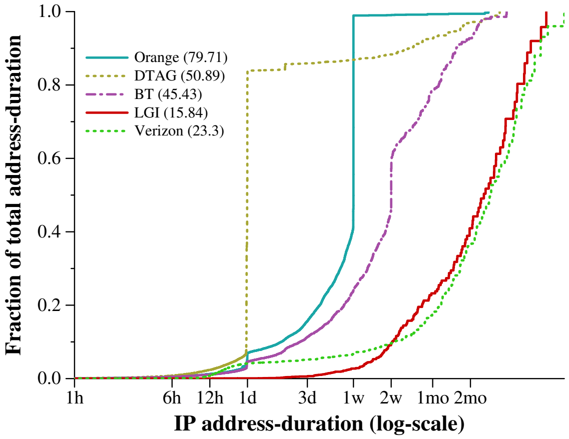


Figure 4.2: Cumulative distribution of total time fractions for ASes with most RIPE Atlas probes that yielded at least one address duration. Probes from Orange and DTAG spent more than half of their total duration in periodic durations of 1 week and 1 day respectively. BT also showed evidence of periodic renumbering with a mode at two weeks. On the other hand, LGI and Verizon have no modes at any durations, and spent most of their total time in durations that were weeks long.

could also simply reflect ISP policy, perhaps to hinder users from running web servers as anecdotal evidence suggests [98]. Investigating AS-level behavior can inform whether the periodic renumbering behavior is concentrated in some ASes and absent in others, shedding light on its potential cause.

4.6.3.1 Is periodic renumbering prevalent across all ISPs?

We first investigate the ASes with the largest deployment of RIPE Atlas probes where we detected at least two instances of address changes. Recall that we only obtain an address duration when the address began and ended during the interval we studied, so that a minimum of two address changes are necessary for a probe to yield an address duration. Figure 4.2 shows the cumulative distribution of total time fractions for the five autonomous systems with the most probes that yielded address durations. In this figure, Orange, an ISP from France, appears to change addresses after a duration of 168 hours (1 week): 55% of its total address duration was a week long. The German ISP, Deutsche Telekom AG (DTAG) reassigns addresses after 24 hours: 76% of the total address duration lies in that mode. British Telecom (BT) has a mode at 336 hours (2 weeks) with 13% of its total duration being in 2 week intervals. We study these ASes further in Section 4.6.4.

The other two ISPs do not exhibit any evidence of periodic renumbering. Liberty Global, an ISP to which probes spread across Europe belong, does not appear to change addresses periodically and neither does Verizon (US). Among these ASes, Verizon has the longest address durations.

Since periodic renumbering behavior is widespread in some ISPs and nonexistent in others, we conclude that the cause of periodic renumbering is likely ISP policy.

4.6.3.2 Is periodic renumbering geographically correlated?

Next, we investigate how the periodic renumbering behavior of ISPs correlates with the country in which they operate. Germany has more than a hundred RIPE Atlas probes deployed across several ISPs, thus we study their address durations in Figure 4.3 for ISPs with probes that contributed at least 3 years of total time. Many ISPs in Germany change addresses every 24 hours: 77% of the du-

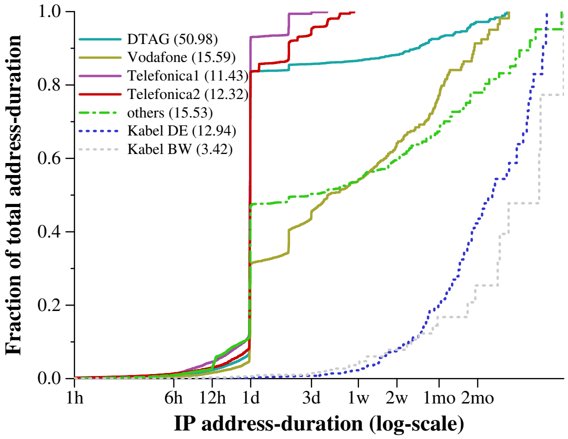


Figure 4.3: Cumulative distribution of total time fractions for ASes in Germany. Many German ISPs appear to change addresses every 24 hours. However, some ISPs have more stable addresses.

ration in DTAG (AS 3320), 76% in Telefonica1 (AS 6805), 74% in Telefonica2 (AS 13184), and 29% in Vodafone (AS 3209), is 24 hours. We observe that the 'other' ISPs also have a mode at 24 hours, suggesting that German ISPs are particularly likely to renumber every 24 hours. However, this behavior is not universal: Kabel Deutschland (AS 31334) and Kabel BW (AS29562) do not exhibit a mode at 24 hours; instead, more than 90% of their total address duration was spent in durations longer than two weeks.

These results suggest that periodic renumbering behavior can exhibit some geographic correlation, but is likely largely caused by ISP policy.

Private communication with a large European ISP confirmed that the ISP renumbers every 24 hours, since the ISP considers this scheme to be more 'privacy secure' although there is no government regulation that forces this feature. The

ISP also reported that it uses PPPoE instead of DHCP for its DSL lines (which accounted for the vast majority of its customers). Since periodic behavior would be atypical of DHCP but consistent with PPP techniques for address assignment, we speculate that periodic renumbering is a property of ISPs that use PPP.

4.6.4 ISPs that renumber periodically

In this section, we look specifically at ISPs that renumber periodically to infer the period over which they renumber, the fraction of the ISPs' probes which periodically renumber, how reliably the renumbering occurs at the end of the period, and whether the renumbering is synchronized across probes. We classify a probe as "periodic" when its total time fraction at some duration d exceeds 0.25. We set the threshold to 0.25 because we expect a probe whose address is reassigned periodically to sometimes have a shorter duration, say, due to a reboot, and sometimes have a longer duration, say, by receiving the same address again.

We consider autonomous systems having at least five probes with an address change of which at least three probes are periodic, and provide an overview of their renumbering period and behavior in Table 4.5. The periodic duration d is shown in hours; 24 hour durations are typical. Renumbering in this table is primarily a feature of central Europe, with some in Russia, Kazakhstan, Mauritius, and South America. We describe the rest of the columns in the next subsections.

4.6.4.1 What fraction of probes is periodic?

Even for ISPs such as Orange and DTAG which have total time fraction at period d in excess of 0.5, not all address durations equal d; some durations are shorter and others longer as seen in Figure 4.2. One possible explanation is that only a few probes in these ISPs were periodically renumbered while others were not. Alternately, periodic probes sometimes have address durations not equal to d. We find that it is usually a combination of both factors that lead to non-periodic durations in these ISPs, although the extent to which each is responsible varies by ISP.

In Table 4.5, the N column shows the number of probes with at least one address change in the dataset. The next column, $f_d^p > 0.25$, shows the number of periodic probes—those having a time fraction of more than 0.25 at duration d. In some ISPs, only a small fraction of probes are periodically renumbered. For example, only a fifth of the probes in BT were periodic with a 2-week period, partially explaining why the total time fraction at 2-weeks for BT in Figure 4.2 is only 0.13.

The subsequent columns, $f_d^p > 0.5$ and $f_d^p > 0.75$ show what percentage of the periodic probes are persistently so, where the total time fraction at duration d is more than half or three quarters. We show percentages rather than raw counts in these columns to simplify the comparison, given that these providers have different sizes. A high percentage indicates that most of the periodic probes (with $f_d^p > 0.25$), are strongly so ($f_d^p > 0.75$). A low percentage indicates that probes may either be reassigned early (due to outages) or late (due to inconsistent reassignment). We can see that only 15% of the periodic probes in BT had $f_d^p > 0.5$ and none had $f_d^p > 0.75$, providing further explanation for why the total time fraction at 2-weeks for BT is low.

Other ISPs have a much larger fraction of their probes that are periodic: more than 80% of probes in Orange, DTAG, Telefonica Germany, A1 Telekom, Hrvatski, ISKON, ANTEL, Global Village Telecom, Mauritius Telekom, Orange Polska, and Digi Tavkozlesi are periodic. For each of these ISPs, more than 75% of probes are persistently periodic, having $f_d^p > 0.5$. For DTAG, Telefonica, A1 Telekom, Hrvatski, ANTEL, and Orange Polska, more than 75% of probes have $f_d^p > 0.75$. Notable is Orange Polska, which has four of its ten probes periodic at 24 hours, and five more probes periodic at 22 hours, but 100% of them have a time fraction at their respective durations greater than 0.75.

Probes in these ISPs typically have address durations capped at d. Address durations can sometimes be shorter—potentially due to outages or reboot/reconnect events as we show in Section 4.7—but can occasionally be larger than d as well. We study these next.

4.6.4.2 Why are some address durations longer than the period?

Some address durations exceed the typical period, d, for an ISP. We would like to determine whether this is a behavior limited to a few probes in the ISP (potentially caused by unusually designed CPE devices), or if the longer-than-typical durations are spread across probes.

How many periodic probes have an address duration longer than d? We expected that no address duration for such probes would exceed the periodic duration d. That is, if the ISP was renumbering a probe on a schedule, then some additional renumbering would be possible due to other reasons, but the probe would never keep its address longer than d. It turns out that this expectation is not the case. The column $MAX \leq d$ shows the percentage of the periodic probes that had their maximum address duration less than d (to capture only those durations that clearly exceeded d, we adjusted d to be $d + 5\%$ for this column). Across all periodic probes, 94% of those that appear to be on a one-week renumbering schedule did not have an address duration longer than one week; only 44% of those that appeared to be on a one-day renumbering schedule had all durations limited by twenty-four hours.

This fraction seemed surprisingly low. Why would so many probes show daily renumbering, even reporting a total time fraction of 0.75, when the probe might also keep its address longer? We considered two possible explanations that would have the same symptoms: that a periodic renumbering was skipped or that the same address was (perhaps by random chance) assigned again. In these cases, rather than see an address change after 24 hours, we might see one at 48 or even 72 hours. We term such address changes “Harmonics”, and consider what fraction of the time all address changes are at or before d (as expected), or occur at a multiple of d. The percentage of probes that match this loosened definition (a superset of those in $MAX \leq d$) appears in the last column of Table 4.5. Most periodic probes from all ISPs except Global Village Telecom and SONATEL-AS have maximum durations of this kind.

AS	ASN	Country	d	N	$f_d^p > 0.25$	$f_d^p > 0.5$	$f_d^p > 0.75$	$MAX \le d$	Harmonic
All			24	2272	193	88.6%	68.4%	43.5%	89.6%
All			168	2272	123	74.0%	13.8%	94.3%	98.4%
Orange	3215	France	168	122	111	77%	14%	98%	99%
DTAG	3320	Germany	24	63	51	96%	86%	78%	98%
Telefonica DE 2	6805	Germany	24	17	15	93%	80%	27%	93%
Telefonica DE 1	13184	Germany	24	14	14	93%	86%	21%	100%
PJSC Rostelecom	8997	Russia	24	22	13	100%	69%	23%	100%
BT	2856	U.K.	337	67	13	15%	0%	38%	62%
Proximus	5432	Belgium	36	41	12	83%	8%	0%	83%
A1 Telekom	8447	Austria	24	12	11	100%	91%	73%	100%
Vodafone GmbH	3209	Germany	24	21	9	78%	11%	0%	89%
Hrvatski	5391	Croatia	24	7	7	100%	100%	43%	86%
ISKON	13046	Croatia	24	6	6	83%	33%	0%	100%
ANTEL	6057	Uruguay	12	6	6	100%	100%	33%	100%
Global Village Telecom	18881	Brazil	48	6	6	100%	67%	0%	17%
Mauritius Telecom	23889	Mauritius	24	6	5	100%	20%	20%	100%
JSC Kazakhtelecom	9198	Kazakhstan	24	15	5	80%	80%	60%	80%
Orange Polska	5617	Poland	22	10	5	100%	100%	60%	80%
VIPnet	31012	Croatia	92	7	4	75%	0%	75%	75%
Proximus	5432	Belgium	24	41	4	50%	25%	0%	75%
Digi Tavkozlesi	20845	Hungary	168	4	4	100%	25%	100%	100%
Orange Polska	5617	Poland	24	10	4	100%	100%	50%	100%
Free SAS	12322	France	24	12	3	100%	67%	0%	67%
SONATEL-AS	8346	Europe	24	7	3	33%	33%	33%	33%
Net by Net	12714	Russia	47	7	3	100%	100%	67%	100%

Table 4.5: Autonomous systems that had at least three probes with a total time fraction for duration d (in hours) greater than 0.25. $f_d^p > 0.25$ shows the number of probes that had a total time fraction at d greater than 0.25; $f_d^p > 0.50$ and $f_d^p > 0.75$ show the percentage of those probes that had fractions greater than 0.5 and 0.75 for the same duration. $MAX \le d$ shows the percentage of probes whose maximum duration was no greater than d. "Harmonic" represents the percentage of probes that, if not renumbered after d, are renumbered after some multiple of d hours. The ASes are sorted in decreasing order of $f_d^p > 0.25$.

4.6.4.3 Are changes synchronized?

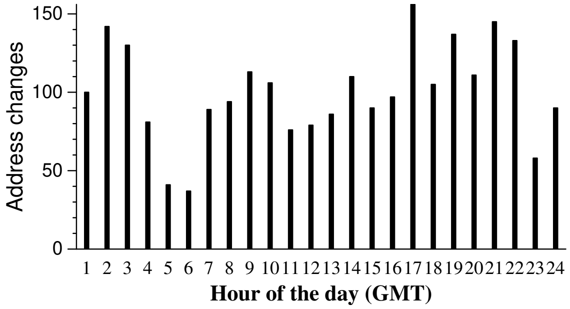


Figure 4.4: Periodic address changes in Orange appear more evenly distributed among the hours of the day.

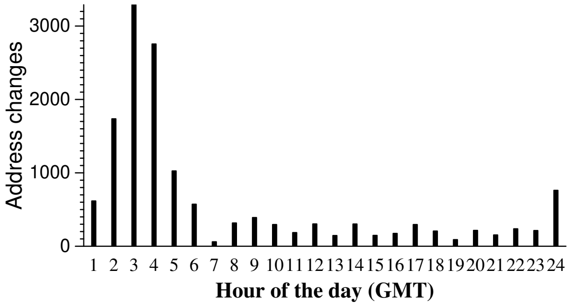


Figure 4.5: Periodic address changes are more likely in some hours for Deutsche Telekom.

We imagine two broad strategies for daily renumbering: either leaving each customer on an independent, free-running clock that resets after 24 hours, or synchronizing all address changes to an off-peak time when few would be interrupted. Both seem reasonable strategies: independent clocks seem simple to implement, synchronized address changes seem more likely to shuffle addresses since many addresses are made available during the synchronized interval. However, if one were to blacklist addresses for misbehavior, knowing which strategy is in use would help to choose for how long to keep the blacklist entry. We expect that plotting the time of day at which addresses change for each ISP will expose whether the renumbering is synchronized.

For Orange and DTAG, the two ISPs with the most periodic probes, we choose the hour of the day in which every address duration that had duration d ended and show these in Figure 4.4 and Figure 4.5. For Orange, periodic address changes are not concentrated during any specific hours of the day. However, DTAG assigns periodic durations more often during some hours of the day. In private correspondence with a large European ISP, we learned that many CPE devices come with an option to choose the time at which they should disconnect and reconnect to receive a new address, as a privacy feature. Figure 4.5 supports this deployment scenario, observing almost three quarters of all periodic address changes between hours 24 to 6 (in GMT). However, some CPEs do not have this feature because a quarter of the periodic address changes happen at other hours of the day.

4.7 Outage-caused address changes

In Section 4.6.4, we saw that even probes from ISPs that renumber periodically often have durations shorter than the typical period. In this section, we study another potential cause of address change: outages occurring at the CPE (customer premises equipment), due to loss of power or network connectivity. Here, we quantify how frequently and for which probes an outage event at the CPE device appears to cause the reassignment of its IP address. If an outage event occurs at approximately the same time as an address change, we assume that the outage caused the address change. If an outage event occurs distant in time from an address change, then we assume that the outage did not cause an address change.

There are three versions of RIPE Atlas probes: v1,v2, and v3. More than 75% of probes are v3, although the distribution of versions within individual ISPs varies. We find network outage events on all versions of probes since network outages are by definition caused when a probe was up and reporting measurements. However, finding power outage events is confounded by the presence of potential false positives and negatives. We address these in detail next and describe our approach for filtering falsely inferred power outages.

4.7.1 Filtering falsely inferred power outages

The SOS-uptime data (Section 4.5.5) allows us to determine when the *probe rebooted*. Ideally, however, we would like to know *when the CPE rebooted*. Fortunately, probe reboots are often representative of CPE reboots due to a combination of how the RIPE NCC suggests that probes be installed [99] and expected fate sharing of co-located devices powered together, as we describe next.

The RIPE Atlas probe gets power from USB; because of this design, the probe can be powered by the USB port on the CPE and will be power-cycled whenever the CPE reboots. When the probe is plugged into the CPE, or both together are power-cycled, a probe reboot indicates that the CPE also rebooted. These represent the typical cases that are useful for the analysis of power outage related address changes. The potential error scenarios are as follows. When the CPE alone is rebooted but the probe is not, we would not observe a power outage, leading to a false negative. When the probe alone is rebooted but the CPE is not, we would detect a power outage, leading to a false positive. Although we expect probe reboots to be rare, a specific scenario in which they occur is when the probe receives a firmware upgrade. We discuss how to remove probe reboots due to firmware upgrades below in Section 4.7.2.

Older probe hardware (v1,v2) can also confound our inference of power outages, because these probes may reboot when they create new TCP connections, since they are vulnerable to memory fragmentation [100]. Address changes create new TCP connections and could induce such reboots, so for our power outage analysis we discard data from these older probes.

4.7.2 Removing reboots caused by firmware updates

The RIPE Atlas servers push firmware updates to probes simultaneously. When a probe's TCP connection to the central controller breaks, the probe will reboot and install the firmware update. Our goal is to filter reboots that were associated with a firmware update, since these reboots occur *as a result of* a dropped connection rather than as a cause. Figure 4.6 shows the number of unique probes that rebooted on each day of 2015. We observe five periods during the year when probes experienced more than twice as many reboots as the median for at least two consecutive days.

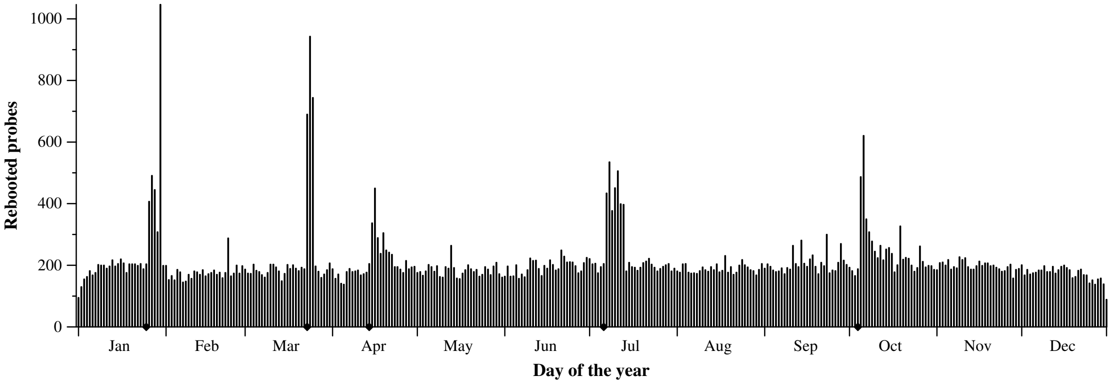


Figure 4.6: Number of unique probes that rebooted on each day of the year. Days with exceptionally many reboots follow the distribution of firmware updates. We indicate days where updates seem to have been distributed with diamonds along the x-axis.

For each of these periods, we found the first day corresponding to the spike, and identify that day as when the firmware update was distributed. Some dates (April 14, July 6, October 5), agree precisely with documented RIPE Atlas firmware and UI updates [101]. Other dates are close—we observe March 23 instead of March 28, and January 25 instead of January 14—but nevertheless show the same spike in reboots. We then discard the first reboot for each probe that occurred after the firmware update.

4.7.3 Most outages result in an address change for some ASes

We found network and power outage events and associated them with inter-connection gaps as described in Section 4.5. If the connection log entries on either side of the inter-connection gap used different addresses, we infer that the event caused an address change and call the address change an *Address change with net-*

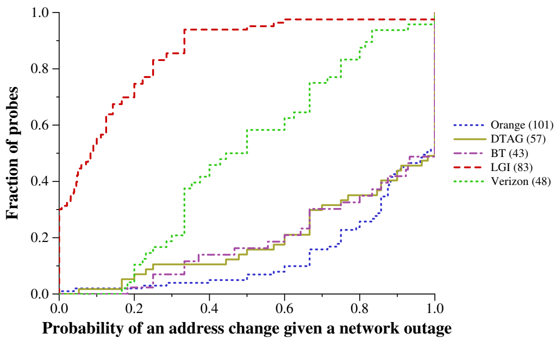


Figure 4.7: Distribution of $P(ac|nw)$ per probe for the ASes with the most probes that had at least one address change. Probes in DTAG, Orange, and BT, are far more likely to change addresses upon a network outage than probes in Verizon and LGI.

work outage, *Address change with power outage*, and *Address change with no-outage*, depending upon the event.

For each individual probe, we consider the conditional probability of an address change given a detected outage. $P(ac|nw)$ represents the conditional probability that an address change occurred given a network outage and $P(ac|pw)$ represents the same for a power outage. We estimate this probability using the fraction of outages occurring contemporaneously with an address change (out of the total number of outages). We show the distribution of these probabilities by probe to estimate whether the group of probes (by geography or ISP) is dominated by those that always or seldom change addresses on an outage.

We find that the likelihood of address change upon an outage event differs across ASes. Figure 4.7 shows the CDF of $P(ac|nw)$ for the five ASes that host the most probes with at least one address change and at least three network outage events. We find that probes in ASes that periodically renumber—Orange, DTAG,

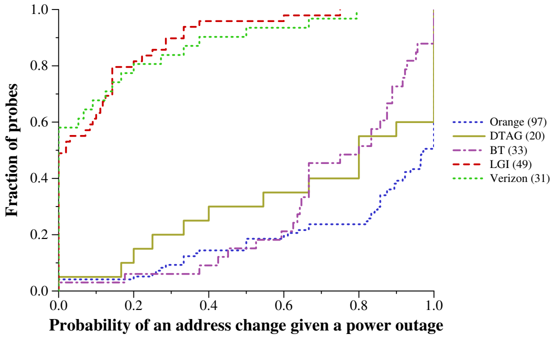


Figure 4.8: Distribution of $P(ac|pw)$ per probe for probes running version 3. As with network outages, probes in DTAG and Orange are more likely to change addresses upon power outage than probes in Verizon and LGI.

and BT—have high $P(ac|nw)$ compared to probes from ASes that do not periodically renumber, LGI and Verizon. Around half of the probes in both Orange and DTAG had $P(ac|nw)$ equal to 1: every network outage was accompanied by an address change!

Figure 4.8 shows $P(ac|pw)$ for these ASes. Recall that we discarded probes with versions 1 and 2 due to their potential to reboot as a result of an address change, thus we have fewer samples. The AS-level behavior for power outages is similar to network outages. DTAG and Orange tend to renumber frequently upon power outages; half of the probes in Orange and 40% of the probes in DTAG have $P(ac|pw)$ equal to 1. Verizon and LGI do not renumber frequently upon power outages; only about half of their probes had an address change even once upon an outage. Since the likelihood of an address change upon an outage can also depend upon the duration of the outage, we investigate the distribution of outage

AS	ASN	Country	N	$P(ac\|nw) > 0.8$	$P(ac\|nw) = 1$	$P(ac\|pw) > 0.8$	$P(ac\|pw) = 1$
All			1113	29.1%	16.9%	28.3%	14.6%
Orange	3215	France	84	79%	54%	77%	50%
Telecom Italia	3269	Italy	28	71%	50%	57%	21%
BT	2856	U.K.	22	64%	55%	50%	14%
Proximus	5432	Belgium	20	70%	45%	60%	30%
DTAG	3320	Germany	19	58%	47%	47%	42%
Vodafone GmbH	3209	Germany	12	83%	75%	58%	42%
Wind Telecomunicazioni	1267	Italy	12	67%	42%	83%	42%
SFR	15557	France	16	38%	25%	50%	6%
ISKON	13046	Croatia	6	100%	50%	83%	67%
PJSC Rostelecom	8997	Russia	7	71%	29%	57%	14%

Table 4.6: Probes likely to change addresses upon network outages are also likely to change addresses upon power outages. The table shows autonomous systems with at least five probes whose conditional probability of address change upon network outage was greater than 0.8. The N column shows the number of probes with at least three network outages and at least three power outages. $P(ac|nw) > 0.8$ and $P(ac|nw) = 1$ show the percentage of N for which the conditional probability of address change upon network outage was greater than 0.8 and equal to 1 respectively, and $P(ac|pw) > 0.8$, $P(ac|pw) = 1$ show the same for power outages.

durations and the likelihood of address changes for different outage durations in Section 4.7.4.

Since the ASes in Figure 4.7 and Figure 4.8 exhibit such disparate behavior, we considered if some ASes are particularly likely to renumber upon outages. To investigate this, we found the set of probes with at least three network and power outages. We then found probes with $P(ac|nw)$ of 0.8 or more and show ASes with 5 or more such probes in Table 4.6.

First, we observe strong geographic correlation; all these ISPs are in Europe. Second, we observe that $P(ac|pw)$ is also high; $P(ac|nw) > 0.8$ and $P(ac|pw) > 0.8$ are similar for all these ISPs (although $P(ac|pw) = 1$ tends to be lower because our power outage detection technique is more prone to false positives). This suggests that both types of outages are likely to cause address changes. Third, we find that 7 of the 10 ISPs also appeared in Table 4.5. Maier et al. [87] studied the logs from an urban area of a major European ISP that used Radius to assign addresses: neither CPE nor Radius servers remember addresses. The behavior of these ISPs that nearly always renumber is consistent with the behavior of the large DSL provider in that study. Private communication with a large European ISP whose probes consistently had an address change upon outage confirmed that they use PPPoE and Radius to assign addresses for their DSL lines. We expect that this property can be used as evidence in inferring a device's link type.

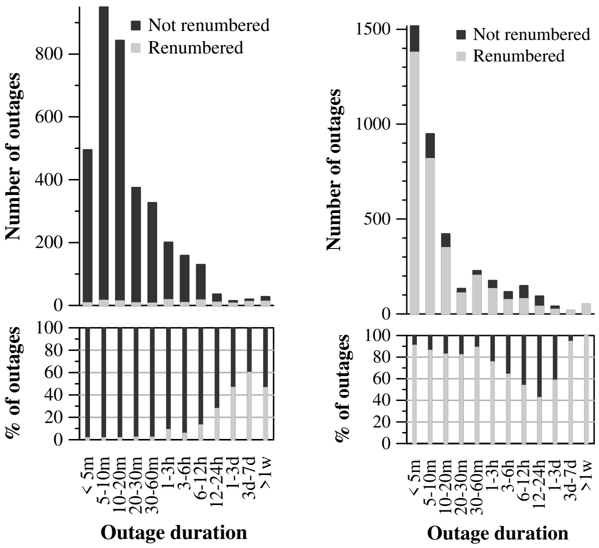


Figure 4.9: The likelihood of an address change (renumbering) given network or power outages of different durations in LGI (left) and Orange (right). The top graph is a histogram; the complete bar represents the number of outages observed across all probes in that AS. The lightly-shaded bar extends for those outages that also saw an address change. The lower graph shows the same data as a percentage. Although relatively few outages lasted longer than a day, the majority of these were coincident with an address change in both ISPs. However, Orange (right) changed addresses even on the shortest outages.

4.7.4 Is there a relationship between outage duration and address changes?

Dynamic addresses assigned using DHCP should typically retain their addresses as long as they continue to renew their lease half-way into the lease duration as the standard recommends [79]. However, an outage could prevent them from renewing their lease. Depending upon the address churn at the time, the address they had previously been assigned may be reassigned to another device. In this way, an outage longer than half a lease duration could potentially cause an address change.

To investigate this, we analyzed the conditional probability of an address change given the occurrence of network or power outages of different durations for probes from LGI (AS 6830) and Orange (AS 3215) in Figure 4.9. For network outages, we considered outages from all versions of probes while for power outages, we only considered outages from probes running v3. We chose these ISPs due to their difference in address change behavior upon the occurrence of outages as seen in Figure 4.7 and Figure 4.8.

The behavior upon outages for the two ISPs is strikingly different. LGI's behavior appears consistent with what we would expect for dynamic addresses assigned using DHCP: fewer than 3% of outages of up to an hour resulted in an address change. More than 25% of outage durations that lasted at least twelve hours resulted in an address change. This behavior is consistent with a DHCP lease duration on the order of a few hours. Not every outage longer than twelve hours resulted in an address change, consistent with DHCP behavior when a client returns after an expired lease and the previously assigned address is still available.

For Orange, we found that even very short outages resulted in address changes. 91% of outages that lasted less than five minutes resulted in an address change, and for every outage duration longer than five minutes and shorter than three hours, more than 75% occurred with an address change. For outages between three hours to three days long, the percentage of address changes was closer to 50%, suggesting the presence of some CPE devices that do not renumber upon every outage. However, as the outage duration increases beyond 3 days, almost every outage results in an address change.

Private communication with a large European ISP confirmed that this behavior is expected for PPPoE based DSL lines in that ISP: any reboot/reconnect event will result in the assignment of a new address from the ISP's dynamic address pool. Since outages of such short durations can result in an address change, a simple reboot of the CPE (resulting in a power outage), or unplugging and replugging the network cable (resulting in a network outage), can change the dynamic address assigned to the end-user. That end-users can change their dynamically assigned address has implications for researchers and operators who use IP addresses to identify end-hosts, particularly when IP addresses are being used to blacklist malicious actors.

4.8 Does a user's dynamic address prefix change?

It is tempting to expect that a new address, when reassigned, will typically be drawn from nearby addresses, say, from the same enclosing /24 prefix. If such an assumption were true, it would allow blacklisting of the enclosing prefix of a malicious host, if it were thought that the malicious host could cause its address to change via reboot or by waiting a day. However, we find that such locality of addresses is rare and address changes typically span prefixes.

We examined whether the dynamic address assigment also varies the enclosing prefix, defined three ways. For each instance of address change that we observed, we found the BGP prefix of the previous address and the new address using CAIDA's IP-to-AS dataset [96], as described in Section 4.5. We also extracted the /16 and /8 prefixes from the previous and new addresses. We then compared how often the prefix of the previous address differed from the prefix of the new address. Table 4.7 presents the results for the overall AS-level dataset with 2,272 probes and for the ten ASes with the most probes that had at least one address change.

ISPs varied prefixes even for consecutive addresses assigned to the same customer; nearly half of the 166,644 total address changes we observed also changed BGP prefixes. Unlike periodicity and renumbering upon outages, assigning addresses out of different prefixes appears to be a common behavior for ISPs. For the ten ASes in Table 4.7, Verizon and DTAG had the lowest percentage of address changes across prefixes, but even for these ASes, almost a quarter of all address changes were across /16s and a fifth of all address changes were across /8s. Thus, it is not just the dynamic addresses that change; their prefixes change too. When a malicious actor receives a new address, even blacklisting the entire enclosing /8 prefix of the old address would fail to prevent access for a third of the address changes we observed.

AS	ASN	Country	Diff BGP		Diff /16		Diff /8	
	All		81,571	48.9%	79,430	47.7%	55,835	33.5%
Orange	3215	France	7,016	68%	6,961	67%	5,513	53%
LGI	6830	many	171	56%	168	55%	136	45%
BT	2856	U.K.	1,736	44%	2,685	68%	1,735	44%
DTAG	3320	Germany	4,706	24%	5,391	28%	4,610	24%
Verizon	701	U.S.	241	23%	241	23%	209	20%
Comcast	7922	U.S.	76	37%	74	36%	63	31%
Proximus	5432	Belgium	2,152	49%	2,331	53%	1,983	45%
Telecom Italia	3269	Italy	4,281	85%	4,412	88%	2,374	47%
Ziggo	9143	Netherlands	18	35%	22	43%	16	31%
Virgin Media	5089	U.K.	46	84%	49	89%	39	71%

Table 4.7: Number of address changes across prefixes. Diff BGP shows the number of address changes where the previous address and the next address belonged to different BGP prefixes. Diff /16 shows the number of address changes where the previous address and the next address belonged to different /16 prefixes and Diff /8 shows the number of address changes where the previous address and the next address belonged to different /8 prefixes. The % column shows the percentage of total address changes for that autonomous system.

4.9 Using complementary datasets that provide IDs to confirm outages

The results from the RIPE Atlas measurement study allow the identification of networks with stable addresses. However, they also show the existence of networks where dynamic addressing is common. In this section, I show how to use a complementary dataset to confirm outages detected in networks where dynamic addressing can occur.

Recall that Section 4.2 had described the different ways probing-based techniques can make false inferences about outages when dynamic addressing occurs. If the address being probed is withdrawn from its home router and not reassigned to any other device, probe responses will cease to arrive and a false outage will be inferred. If the home router experiences an outage causing its address to cease to respond to probes, and before the outage ends, the address is reassigned to some other device which responds to probes, probing-based techniques will infer the occurrence of the outage correctly but will falsely conclude that the outage ended before it did.

When a responsive address being probed is withdrawn from its host or when the host it belongs to experiences an Internet outage, a probing-based technique will observe that responses cease to arrive. I define this event to be a dropout. Formally: *A dropout happens when the address attached to a residential link transitions from being responsive to pings from multiple vantage points, to being unre-*

sponsive from all of the vantage points. An observed dropout can either be due to an outage or dynamic reassignment.

My key insight is that a complementary dataset which can yield some sort of an unchanging identifier (an ID) uniquely associated with the device can provide information about whether the device's address changed. For instance, consider the probe-ID field provided by RIPE Atlas, which uniquely identifies a device. If the address associated with the device before and after the dropout is the same, it is proof that dynamic address reassignment did not occur. The only way that the address before and after the dropout can be identical and yet for dynamic reassignment to have occurred, is if the device's address changed to a new one and then changed back to the original address. However, Section 4.8 showed that subsequent addresses are often assigned from entirely different prefixes; thus, the probability that a subsequent address is exactly the address that was assigned before is small. Since dynamic reassignment is highly unlikely to have occurred, we can infer that an outage occurred.

The ID-based approach provides two benefits:

- It can offer confirmation of the occurrence of an outage.
- It allows the estimation of outage recovery durations for the instances where an outage is confirmed.

4.9.1 CDN dataset provides IDs

I measure how often addresses remained the same before and after a dropout using a dataset of CDN software logs that contain a timestamp, unique identifier of the software installation on the client machine and the public source IP address visible to the CDN. The CDN offers a service 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. The CDN records logs collected from its software installations on users' desktops and laptop machines. Each logline contains (among other fields) the timestamp at whch the logline was created, the unique identifier of the software installation on the machine (the ID), and the public IP address seen by the CDN's infrastructure at this time. Loglines in the CDN software dataset are dependent on user activity, and therefore, their frequency varies.

4.9.2 Confirming outages detected by Thunderping

For dropouts detected by the Thunderping system [5], I measure how frequently the complementary dataset confirmed that the dropout was an outage. I used all dropout events detected in three years (2015, 2016, 2017) in this analysis and compare against the CDN dataset during the same period.

To determine whether the address associated with a home router remained the same before and after a detected dropout, I first collect all entries where the address that experienced the dropout is present in the log up to one week before the start time; this applies to only about one percent of the dropouts. The matched address is ip_p (for previous), and I refer to the next address after the dropout ip_n (for next). There are three categories of comparison that I show in Table 4.8:

1. When $ip_p = ip_n$, there was no apparent reassignment, which suggests that an outage occurred and that an inferred outage duration is correct.

2. When $ip_p \neq ip_n$, and the observation of ip_n was before ip_p became responsive again, the address was reassigned and the inferred outage duration is incorrect. An outage may or may not have occurred.

3. When $ip_p \neq ip_n$, and the observation of ip_n was after ip_p became responsive again, the address was reassigned but the address change may be independent. Again, an outage may or may not have occurred.

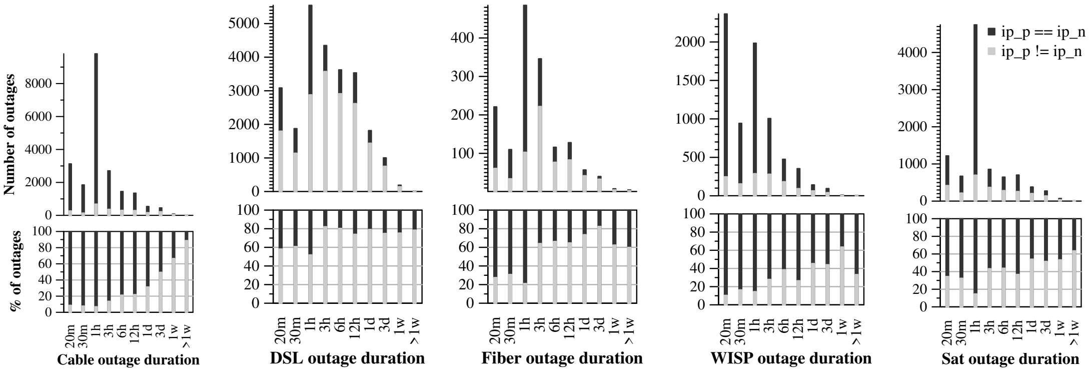
Table 4.8 shows that 60% of Thunderping's detected dropouts when considering all linktypes are *not* accompanied by address changes; thus the majority of dropouts are outages. Additionally, the table shows that nearly all dropouts for addresses with cable connections are outages, corroborating the results from RIPE Atlas which suggested that cable addresses tend to be stable.

For DSL addresses, 31% of dropouts were confirmed. Without the complementary dataset, all of these dropouts were suspect, since prior results showed that DSL addresses tend to be renumbered frequently. However, through the use of a dataset that provides IDs, I am able to confirm outages even in networks where addresses are not stable.

			$ip_p \neq ip_n$	
Link Type	ip_p **present**	$ip_p = ip_n$	**during**	**after**
ALL	84837 (0.7%)	50973 (60.1%)	4765 (5.6%)	29047 (34.3%)
CABLE	21455 (1.1%)	18860 (88.0%)	354 (1.7%)	2221 (10.4%)
DSL	25061 (0.9%)	7761 (31.0%)	2857 (11.4%)	14422 (57.6%)
FIBER	1516 (1.0%)	853 (56.3%)	60 (4.0%)	603 (39.8%)
WISP	7381 (1.1%)	6013 (81.5%)	177 (2.4%)	1191 (16.1%)
SAT	9600 (0.4%)	6939 (72.3%)	241 (2.5%)	2412 (25.1%)

Table 4.8: Confirming Thunderping outages across link types: outages are confirmed when $ip_p = ip_n$.

We next try to determine whether longer apparent outages correlate with address changes. If short outages typically have no address change, we can at least characterize short outage durations. However, if all dropouts lead to address changes on recovery, the time until an address starts responding again is more a function of address reuse than of recovery. Figure 4.10 shows the results for each of the media types in our study. This uses the same data as in Table 4.8. In Figure 4.10, the top graphs represent the raw histograms of apparent outage duration, though only the distribution of dark bars (where the address is unchanged) should be taken as a distribution of true outage duration. The bottom graph represents the fraction of outages having an address change or no address change. At a high level, graphs with more dark are media types or durations that are more likely to be true outages rather than address renumbering. For



(a) Cable **(b)** DSL **(c)** Fiber **(d)** WISP **(e)** Satellite

Figure 4.10: Outage duration vs. probability of address change for addresses from various link types.

WISP and Cable, the bulk of the outages at most 3 hours long have very little renumbering and outage durations can be estimated well. For DSL, even short apparent outages are often accompanied by address changes, meaning that outage duration should not be estimated based on responsiveness alone (to do so would require additional data from clients). We observe few Fiber outages, but the time-dependence is more pronounced.

4.10 Conclusion and Discussion

In this chapter, I showed that dynamic address reassignment can confuse probing-based techniques and lead them to make false inferences about outages. Next, I conducted a measurement study with colleagues to infer and analyze patterns of address changes using an existing set of logs from 3,038 globally distributed RIPE Atlas probes that saw address changes in 2015. We found several factors in play. Dynamic address durations vary by geography, with addresses from North American ISPs persisting for weeks and addresses from many German ISPs assigned for a day. Dynamic addresses change as a result of network and power outages in most ISPs. In some ISPs, an outage of any duration results in an address change, while in others, the likelihood of address change increases with outage duration. Using this study, I was able to identify which networks have stable addresses, where dynamic reassignment is uncommon. I also showed using a complementary dataset that it is sometimes possible to confirm probing-based techniques' detected outages even in networks with frequent dynamic reassignment.

Chapter 5: The need for measuring individual address outages

In this chapter, I develop and evaluate an approach to detect dependent *Internet disruption* events that affect multiple residential addresses simultaneously using measurements of individual address disruptions gathered with the Thunderping technique. Borrowing terminology from Richter et al. [12], I define an Internet disruption event for an address to be the abrupt loss of response to active probing from that address.

Techniques that detect outages at the Internet's edge often seek disruption events affecting a substantial set of addresses. The set of addresses may comprise those belonging to the same /24 address block [11, 12], BGP prefix [13], or country [14]. Techniques seek such disruption events because individually, each large disruption has impact and their size makes them easier to confirm, e.g., with operators. In contrast, disruptions affecting only a few users are harder to detect with confidence. For example, the lack of response from a single address might best be explained by a user switching off their home router—hardly an outage. However, residential Internet outages may be limited to a small neighborhood or apartment block; prior techniques are likely to miss such events.

In the rest of this chapter, I describe work with colleagues where we demonstrate a technique that detects disruption events with quantifiable confidence, by investigating the potential dependence between disruptions of multiple IP addresses in a principled way. We apply a simple statistical method to a large dataset of active probing measurements towards residential Internet users in the US. We find times when multiple addresses experience a disruption simultaneously such that they are unlikely to have occurred independently; we call the occurrence of such events *dependent disruptions*. We characterize these dependent disruption events and present results that challenge conventional wisdom on how such disruptions affect Internet address blocks. We show that many of these events would be missed by existing techniques that do not perform individual address outage detection.

5.1 Background: dependent residential outages can be small

Residential Internet connections are vulnerable. The last-mile link connecting home routers to their ISP is typically not multi-homed and is therefore a single point of failure. Further, last-mile links can be damaged by exposure to the elements or by broken tree limbs blown by the wind. Thus, residential outages may be limited to a small neighborhood or apartment block.

5.1.1 Prior techniques focus upon larger disruptions

Prior techniques that detect edge Internet disruptions typically detect disruptions that affect a group of addresses *collectively*. Like us, they also leverage the *dependence* among the per-IP address "disruptions" that these larger disruptions cause. However, they differ from our techique in that they look for dependence in large aggregates (that is, so many addresses are affected at the same time that there must be an evident anomaly) or limit their resolution to small address blocks, looking only for outages that cause dependent disruptions for *all* the addresses in a monitored block. Thus, these techniques may miss observing smaller residential failures.

For example, Trinocular looks only for outages affecting /24 address blocks [11]. Using historical data from the ISI census [47], it models the responsiveness of blocks and finds addresses within each block that are likely to respond to pings. The system pings a few of these addresses from each block at random in 11-minute rounds. Trinocular then employs Bayesian inference to reason about responses from blocks. When a block's responsiveness is lower than expected, Trinocular probes the block at a faster rate and eventually detects an outage when the follow-up probes also suggest the block's lack of Internet connectivity. Since Trinocular will not identify an outage if a single address in a block responds to probing, Trinocular potentially neglects outages affecting /24 blocks only partially, including larger outages affecting multiple /24 blocks.

Other systems have also investigated disruptions affecting entire blocks of addresses. Recently, Richter et al. used CDN logs to detect disruptions affecting /24 address blocks [12]. Hubble detects prefix-level unreachability problems [13]. The IODA system looks for the most impactful outages only, those causing an extensive loss of connectivity for a geographical area or Autonomous System (AS) [14, 37].

Disco [41] shares some features with our work: they also detect simultaneous disconnects of multiple RIPE Atlas probes within an ISP or geographic region to infer outages. However, there are two major differences between the Thunderping and RIPE Atlas datasets. At any given point in time, the Thunderping dataset typically consists of pings sent to roughly 50,000 addresses in relatively small geographical areas with active severe weather alerts. The Disco dataset consists of 10,000 RIPE Atlas probes distributed around the world; this sparse distribution may prevent the detection of smaller outages localized to one area (like a U.S. state). The second difference is that unlike Thunderping ping data whose timestamps are only accurate to minutes, the timestamps available in the RIPE Atlas datasets are accurate to seconds, permitting the use of Kleinberg's burst detection to detect bursts in probe disconnects. Discussions with the authors of Disco suggested that Kleinberg's burst detection model would not be appropriate for the Thunderping data, although a more detailed evaluation of the binomial test against Kleinberg's burst detection in the Thunderping data is future work.

5.1.2 The Thunderping dataset yields per-address disruptions

The key insight behind our technique is that simultaneous disruptions of multiple individual IPv4 addresses could occur due to a common underlying cause. We therefore require per-IP address disruptions.

Such data is present in the Thunderping dataset [5]. Thunderping pings sampled IPv4 addresses from multiple ISPs in geographic areas in the United States. Originally designed to evaluate how weather affects Internet outages, the system uses Planetlab vantage points to ping 100 IPv4 addresses from multiple ISPs in each U.S. county with active weather alerts. Each address is pinged from multiple Planetlab vantage points (at least 3) every 11 minutes, and addresses in a county are pinged six hours before, during, and after a weather alert.

Here, we analyze a dataset of Thunderping's ping responses to detect disruptions for each probed address using Schulman and Spring's technique [5]. When an address that is responsive stops responding to pings from all vantage points that are currently probing it, we detect a disruption for that address. Since a disruption is detected only when all vantage points declare unreachability, the minimum duration of a disruption is 11 minutes (at the end of 11 minutes each vantage point has pinged the address at least once). Thunderping continues to probe an address after it has become unresponsive, allowing us to estimate how long the unresponsive period lasted.

While per-IP address disruptions allow the detection of small disruptions, all per-address disruptions are not necessarily the result of Internet connectivity outages. For example, an individual user may decide to turn off their home router. In the rest of this chapter, we show how to detect dependent disruption events using per-address disruptions.

5.2 Detecting dependent disruptions

In this section, we apply binomial testing to identify dependent disruptions in the outage dataset. First, we show how the binomial test works to rule out independent events and show how to apply the test to network outages in reasonably sized aggregates of addresses. Second, we apply this method to the outage dataset, omitting addresses with excessive baseline loss rates and evaluating our chosen aggregation method. Finally we summarize the dependent disruptions we found in this dataset. This sets up analysis of these events (time of day, geography, and scope) which we defer to the following section.

5.2.1 Finding dependent events in an address aggregate

When many addresses experience a disruption simultaneously, there could be a common underlying cause. Such disruptions are statistically *dependent*. To identify these dependent events, our insight is to model address disruptions as *independent* events; when disruptions co-occur in greater numbers than the independent model can explain, the disruptions must be *dependent*. Binomial testing provides precisely this ability to find events that are highly unlikely to have occurred independently.

Given N addresses, the binomial distribution gives the probability that D of them were disrupted *independently* as:

$$\Pr[D \text{ independent failures}] \;=\; \binom{N}{D} \cdot P_d^D (1 - P_d)^{N-D} \tag{5.1}$$

where P_d represents the probability of disruption for the aggregate N. To apply this formula, we must first set a threshold probability below which we consider the simultaneous disruption to be too unlikely to be independent. We set this threshold to 0.01%. We then solve for D_{min}, the smallest (whole) number of simultaneous disruptions with a smaller than 0.01% chance of occurring independently.

Table 5.1 presents D_{min}, computed for various values of N and P_d. This table shows that, even for large aggregates of IP addresses, often few simultaneous disruptions are necessary to be able to confidently conclude that a dependent disruption has occurred. When applied to the Thunderping dataset, D_{min} values are typically below 8.

There are two practical challenges in applying this test. First, we must choose aggregates of N IP addresses that define the scope of a dependent disruption: too large an aggregate will have too large a chance of simultaneous independent failures and drive up D, while too small an aggregate may fail to include all the addresses in an event. Second, we must estimate P_d for each aggregate. We address each in turn.

N	$\mathbf{D_{min}}$			
	P_d = 1/hour	1/day	1/week	1/month
10	8	3	2	2
50	21	5	3	2
100	35	7	4	3
500	126	14	6	4
1000	231	21	8	5
5000	1021	64	17	8
10000	1980	112	26	11
50000	9491	457	85	29

Table 5.1: D_{min} values for varying values of N and P_d. There is less than 0.01% probability according to the binomial test that D_{min} or more addresses fail for each N and P_d.

5.2.1.1 Choosing aggregate sets of IP addresses

Our technique assumes some *aggregate* set of IP addresses among which to detect a dependent disruption. We note that the *correctness* of our approach does not depend on how this set is chosen—the binomial test will apply so long as independent failures can be modeled by P_d. When applying our technique, IP addresses must be aggregated into sets that are large enough to span interesting disruption events, but not so large as to become insensitive to them.

In this paper, we aggregate IP addresses based on the U.S. state and the ASN they are in. *State-ASN* aggregates have the benefit of spanning multiple prefixes (so we can observe whether more than one /24 is affected by a given disruption event), but also being constrained to a common geographic region (so hosts in an aggregate are likely to share similar infrastructure). There are two limitations with this approach: states are not of uniform size, though the test elegantly handles varying N, and a few ISPs use multiple ASNs, which may hide some dependent failures. Alternate aggregations are possible.

5.2.1.2 Calculating the probability of disruption (P_d)

As a final consideration, we discuss how to estimate the probability of disruption, P_d, from an empirical dataset of disruptions. We assume that the dataset can be separated into a set of discrete "time bins"; this is common with ping-based outage detection, such as Thunderping and Trinocular, which both consider 11-minute bins of time. P_d can be estimated using the following equation:

$$P_d = \frac{\text{\#disruptions}}{\text{\#timebins}} \tag{5.2}$$

Here, #timebins represents the total number of observation intervals used: if a single host was measured across 10 time intervals and five other hosts were all measured across 3, then #timebins = $10 + 3 \cdot 5 = 25$.

We only consider state-ASN aggregates where we were able to obtain a statistically significant value for P_d. For statistical significance, we adhere to the following rule of thumb [102, Chapter 6]: we accept a state-ASN aggregate with t timebins and estimated probability of disruption P_d only if:

$$tP_d(1 - P_d) \geq 10 \tag{5.3}$$

5.2.2 Applying our method to the Thunderping dataset

We investigate all ping responses in the Thunderping dataset from January 1, 2017 to December 31, 2017 and detect disruptions according to the methodology described above. During this time, Thunderping had sent at least 100 pings to 3,577,895 addresses and detected a total of 1,694,125 individual address disruptions affecting 1,193,812 unique addresses. Figure 5.1 shows the top 15 ISPs whose addresses Thunderping had sampled most frequently. These ISPs include large cable providers (Comcast, Charter, Suddenlink), DSL providers (Windstream, Qwest, Centurytel), WISP providers (RISE Broadband), and satellite providers (Viasat).

Filtering lossy addresses

We find that some pinged addresses experience unusually high ping loss rates. These addresses see disruption very frequently, since high loss rates can result in pings from all vantage points to these addresses failing together. Disruptions for such addresses are even more challenging to interpret because a variety of causes can result in high ping loss rates, such as high response latency [19] and ICMP

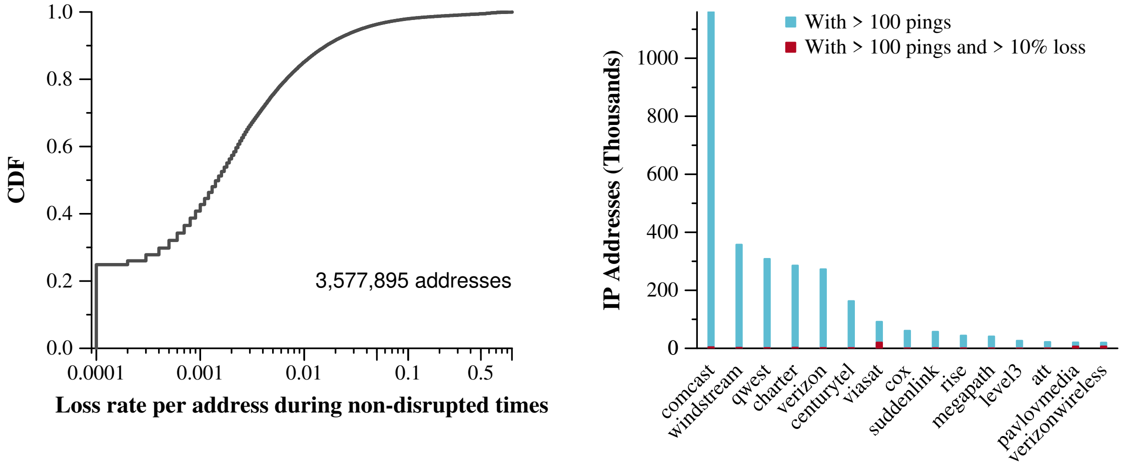


Figure 5.1: (Left) The distribution of ping loss rates per IP address during times when Thunderping believed an address was *not* experiencing a disruption. While most addresses have low loss rates, 2% of addresses had loss rates exceeding 10%. (Right) The fraction of addresses per ISP with ping loss rates exceeding 10% during non-disruption periods, for the 15 ISPs with the most pinged addresses. We filter from all remaining analyses any address whose loss rate exceeded 10%.

rate-limiting [103]. Thus, we find these addresses and remove them from the rest of the analyses.

Figure 5.1 shows the distribution of ping loss rates for IP addresses during times when the addresses were not experiencing a disruption. 2% of addresses have loss rates exceeding 10%. Figure 5.1 shows the prevalence of these addresses in the 15 ISPs whose addresses Thunderping had sampled most frequently. Some ISPs have a higher concentration of addresses with high loss rates, such as Viasat, Verizon Wireless, and Pavlov Media. However, even in these ISPs, the majority of addresses do *not* have high loss rates. Thus, instead of filtering the ISPs whole-

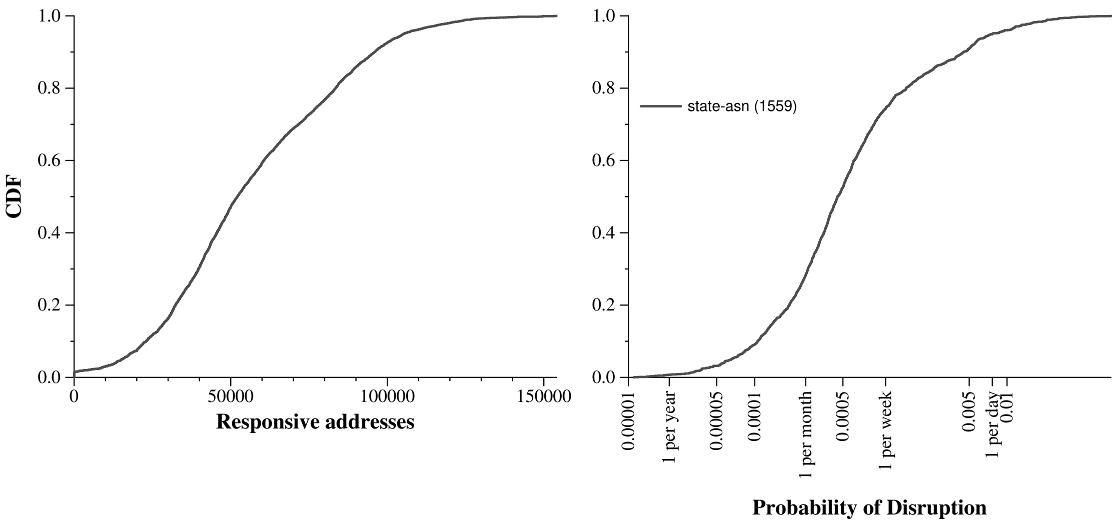


Figure 5.2: Potential N and P_d values in the Thunderping dataset: On the left, we show the distribution of all addresses (across all state-ASN aggregates) pinged by Thunderping that can potentially fail in each 11 minute time bin. On the right, we show the distribution of the probability of disruption (P_d) for various state-ASN address aggregates.

sale, we only remove the addresses whose loss rates exceeded 10% and do not consider these addresses in the remaining analyses.

Detecting dependent disruptions in the Thunderping dataset

We use Figure 5.2 to describe potential N and P_d values in the Thunderping dataset. On the left, we show the distribution of addresses pinged by Thunderping in each 11 minute timebin in 2017. The median number is roughly 50,000 addresses across all U.S. states and ISPs. Since many weather alerts tend to be active at any given point of time, these addresses are likely to be distributed among tens of state-ASN aggregates. In 2017, the maximum addresses that could potentially fail in any state-ASN aggregate was 15,863. On the right, we show the distribution of P_d values for all state-ASN aggregates that we considered. There is extensive variation: addresses in some of these aggregates experience disruptions only once every year, whereas in other aggregates they experience disruptions more often than once per day. [1]

For each state-ASN aggregate, for each 11-minute window during which Thunderping had pinged addresses, we identify the maximum number of addresses that can potentially fail, N, *i.e.,* all the addresses that are responsive to pings at the beginning of the window. Next, we apply the binomial test for each of these windows since we know N and P_d. When the number of disruptions in a window is at least D_{min}, we determine that a dependent disruption event occurred in that window with a probability greater than 0.9999.

N	**Probability of disruption (P_d)**			
	1/hour $> P_d \geq$ 1/day	**1/day $> P_d \geq$ 1/week**	**1/week $> P_d \geq$ 1/month**	**1/month $> P_d$**
≤ 10	11 (0.1%)	486 (2.3%)	519 (2.5%)	179 (0.9%)
≤ 50	6 (0.0%)	1089 (5.2%)	1990 (9.6%)	868 (4.2%)
≤ 100	0 (0.0%)	863 (4.1%)	1229 (5.9%)	736 (3.5%)
≤ 500	0 (0.0%)	1807 (8.7%)	4328 (20.8%)	1360 (6.5%)
≤ 1000	0 (0.0%)	462 (2.2%)	1884 (9.0%)	405 (1.9%)
≤ 5000	0 (0.0%)	171 (0.8%)	1865 (9.0%)	458 (2.2%)
≤ 10000	0 (0.0%)	0 (0.0%)	83 (0.4%)	0 (0.0%)
≤ 50000	0 (0.0%)	0 (0.0%)	32 (0.2%)	0 (0.0%)

Table 5.2: Dependent disruption events for different values of number of addresses that can potentially fail (N) and probability of disruption (P_d) from the Thunderping dataset. Of 20,831 total dependent disruption events, the majority were detected when P_d is low.

[1] Since disruptions are a superset of outages and dynamic reassignment, frequent disruptions are not necessarily indicative of poor Internet connectivity. Also, the existence of many aggregates with few disruptions indicates that Thunderping often pinged addresses during weather conditions that were not conducive to disruptions.

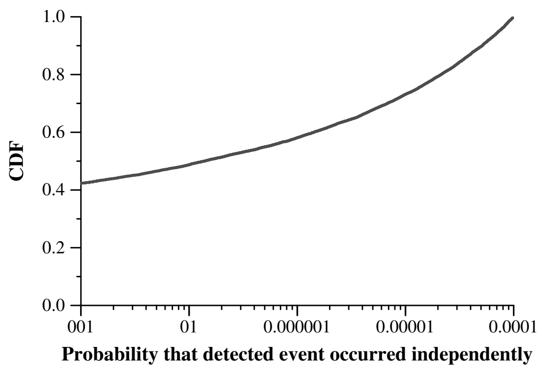


Figure 5.3: Figure 5.3 shows the distribution of the probability that the 20,831 detected dependent disruption events could have occurred independently. For 90% of events, the probability of occurring independently is less than 0.00005.

In total, we detected 20,831 events with dependent disruptions in 2017. Table 5.2 shows the number of detected events for various values of N and P_d in the Thunderping dataset in 2017. The majority of events were detected for state-ASNs with P_d lower than once a week. From Figure 5.2, we know that close to three-quarters of state-ASN aggregates fall in this category, showing that our technique is able to detect dependent disruptions in most aggregates.

Next, we analyzed our confidence in these dependent disruptions. The occurrence of D_{min} disruptions has less than 0.01% probability according to the Binomial test. We test if most detected dependent disruption events have exactly 0.01% probability of occurring or if they are well clear of this threshold.

Figure 5.3 shows the distribution of the probability that we incorrectly classify an independent event as dependent. The probability of occurring independently is less than 0.005% for 90% of the events and less than 0.001% for 75%.

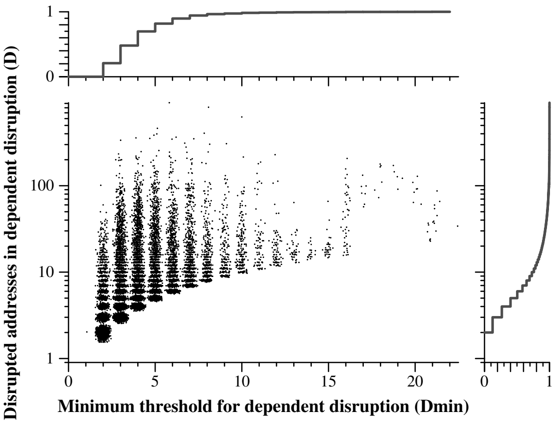


Figure 5.4: For each detected correlated disruption event, Figure 5.4 shows the D_{min} value on the x-axis and the corresponding number of observed disruptions on the y-axis. 62% of the 20,831 detected events had more than D_{min} observed disruptions. The scatterplot adds a random gaussian offset to both x and y with mean of 0.1, clamped at 0.45, to show density.

Thus, the probabillity that detected events occurred independently is typically much smaller than our choice of 0.01%.

How many addresses are disrupted dependently?

The Binomial test does not say that *all* of the addresses that were observed to be disrupted during a dependent event were disrupted in a dependent manner. Consider if D_{min} is 4 and we detect an event where 7 addresses were disrupted. The Binomial Test shows us that the event took place with very low probability. However, that does not necessarily mean all 7 addresses were disrupted in a dependent manner; up to 3 of them could have been disrupted independently with up to 99.99% probability.

We call the set of addresses in a state-ASN aggregate that were disrupted in the time-bin of a dependent event the observed group of addresses that were disrupted, or the *observed disrupted group* for short. Of the observed disrupted group, our assumption is that some were disrupted together in a dependent manner: we call this subset the actual group of addresses that were disrupted, or *actual disrupted group*. We obtain a minimum bound on the actual disrupted group by subtracting $D_{min} - 1$ from the observed disrupted group. For the 20,831 dependent disruption events, the total addresses in all the observed disrupted groups is 229,413 and the total addresses in all the minimum actual groups is 165,328.

We study the relationship between D_{min} for a state-ASN aggregate on the x-axis and the corresponding number of addresses in the observed group of disrupted addresses (on the y-axis) in Figure 5.4. Each point corresponds to one of the 20,831 detected events. Sometimes, a state-ASN aggregate had such low P_d that even a single disruption in a 11-minute bin occurred with less than 0.01% probably and therefore had a D_{min} value of 1. However, since we are looking for unlikely disruptions of multiple addresses, all our detected events observed at least two addresses that were disrupted in the same time-bin.

12,911 (62%) detected events observed *more* than D_{min} disruptions, corroborating the result from Figure 5.3 that most detected events would have been detected even with a stricter threshold.

We detected dependent disruption events with various sizes as shown in Figure 5.4. There are 693 (3%) events with more than 50 observed disrupted addresses. For the largest detected event, we observed 913 addresses experience disruptions in the same time-bin in AS33489 (Comcast) in Florida at 2017-09-13T20:33 UTC time. This detected event correlates to the minute with a known failure event for Comcast that was discussed in the Outages mailing list [104]. However, for most of the events, the size of the observed group of disrupted addresses is small: there were 2,593 (12%) with two, 2,969 (14%) with three, 2,776 (13%) with four, and 2,175 (10%) with five observed disrupted addresses. These results highlight the ability of our technique to detect even small sized disruptions with confidence.

5.3 Properties of dependent disruptions

In this section, we study various properties of dependent disruptions. For some properties, we conduct additional analyses on specific ISPs in the Thunderping dataset: Comcast (cable), Qwest (DSL) and Viasat (Satellite). These are three ISPs whose addresses are pinged frequently by Thunderping (as seen in Figure 5.1) and where we were able to detect in excess of a thousand dependent disruption events (3109 events for Comcast, 1855 for Viasat, 1734 for Qwest).

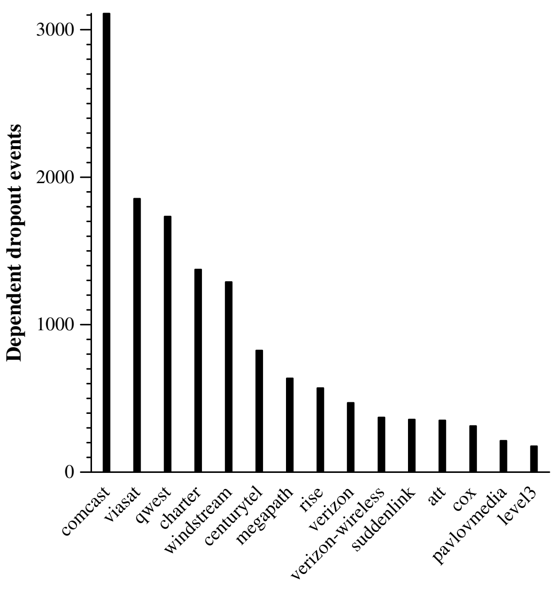
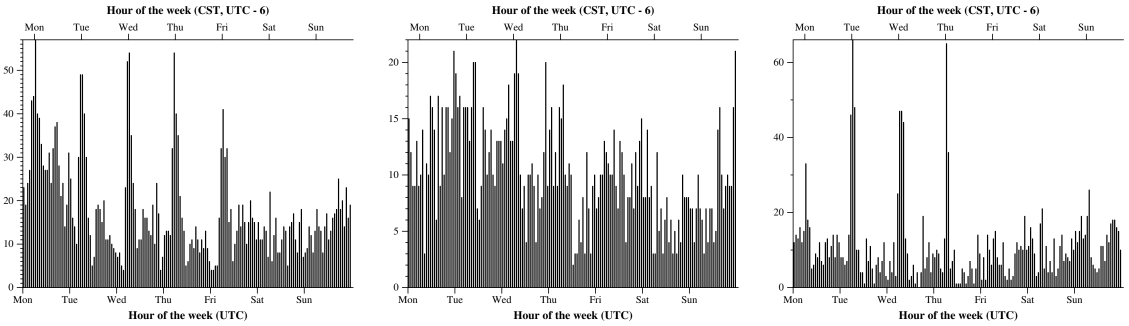


Figure 5.5: Figure 5.5 shows the number of dependent disruption events detected per ISP. Note that these numbers are more a reflection of addresses sampled and pinged in the Thunderping dataset than any major underlying problem in their infrastructure.

5.4 Dependent disruption events across ISPs

We grouped dependent disruption events by ISP to check if any ISPs contribute an unusual number of events. Figure 5.5 shows the top 15 ISPs with dependent disruption events. Most of the ISPs from Figure 5.1 are represented here as well, suggesting that no ISPs are unduly biasing our results. These top 15 ISPs together account for 13,643 (65%) of all detected events.

We emphasize that these results are not meant to reflect any underlying problems with these ISPs; the Thunderping system samples and pings large ISPs more frequently and consequently, finds more disrupted addresses in them. The purpose of this analysis is to ensure that no ISP contributes unduly many events.



(a) Comcast **(b)** Qwest **(c)** Viasat

Figure 5.6: Dependent disruption events that began in each hour of the week. 'Mon' on the bottom x-axis refers to midnight on Monday in UTC time. On the top x-axis, 'Mon' refers to midnight at UTC-6 (CST).

5.4.1 Dependent disruptions are more frequent at night for some ISPs

Recent work has shown that disruptions tend to happen more frequently during maintenance intervals close to midnight local time [12]. To obtain this result, Richter et al. used proprietary data from a content delivery network, collected at the granularity of every hour. Here, we investigate if our technique can identify similar patterns of dependent disruptions.

Figure 5.6 shows that individual ISPs can have different behavior. Comcast and Viasat have more dependent disruption events occurring close to midnight, CST, on weekday nights. Qwest, on the other hand, does not appear to have a clearly discernible pattern. Our results confirm those from prior work [12], lending credence to our technique. Moreover, we are able to do so using public (Thunderping) data and a granularity of every 11 minutes.

5.4.2 Dependent disruptions can recover together

Here, we investigate whether dependent disruption events are accompanied by *dependent recovery*. Since Thunderping continues to probe an IP address even after it becomes unresponsive until the end of the weather alert, it can observe when the address becomes responsive again. This responsiveness may signal that the disruption for the address has ended. Multiple addresses that are disrupted together and also recover together offer evidence that: (a) the event was indeed dependent and (b) the event has ended, allowing estimation of the disruption's duration.

Most dependent disruptions also have correlated recoveries. Of 20,831 dependent disruption events, 6,869 (33%) had *all* disrupted addresses recover during the same 11-minute time-bin. Further, 14,789 (71%) disruption events had at least half of the disrupted addresses recover together. Across all of the 20,831 dependent disruption events, there were 229,413 disrupted addresses in total. Of these, 121,648 (53%) disrupted addresses—from 15,117 (73%) disruption events—exhibited a dependent recovery with other addresses from that same group. This indicates that dependent recovery is quite common.

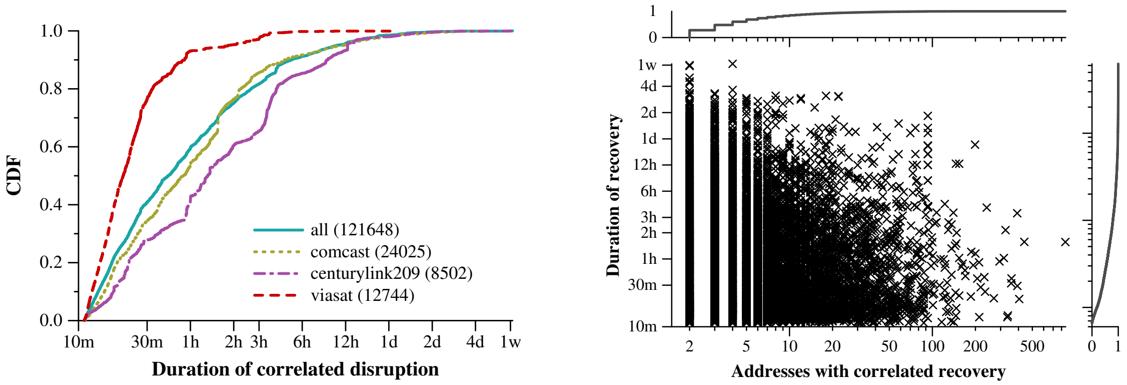
We also tested whether the likelihood of dependent recovery is a function of the number of addresses in the observed disrupted group. It is possible that

Disruptions	Correlated recoveries								
	-2	**-1**	**0**	**2**	**3**	**$\leq$ 10**	**$\leq$ 50**	**$\leq$ 100**	**$\leq$ 1000**
2	623 (24%)	286 (11%)	425 (16%)	1259 (49%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
3	476 (16%)	283 (10%)	463 (16%)	741 (25%)	1006 (34%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
$\leq$ 10	869 (8%)	488 (5%)	1329 (13%)	1938 (19%)	1714 (17%)	3937 (38%)	0 (0%)	0 (0%)	0 (0%)
$\leq$ 50	216 (5%)	78 (2%)	154 (4%)	282 (7%)	281 (6%)	1554 (36%)	1760 (41%)	0 (0%)	0 (0%)
$\leq$ 100	15 (3%)	2 (0%)	4 (1%)	7 (1%)	5 (1%)	54 (11%)	218 (44%)	193 (39%)	0 (0%)
$\leq$ 1000	2 (1%)	0 (0%)	1 (1%)	0 (0%)	0 (0%)	10 (6%)	35 (20%)	52 (30%)	71 (42%)

Table 5.3: The number of addresses that recovered (columns) for dependent disruptions affecting different numbers of addresses (rows). -2 indicates that no addresses that dropped out were observed to have recovered. -1 indicates that only one address recovered. The other numbers show how many of the (at least two) addresses that recovered did so in a correlated manner.

disruptions with fewer addresses in the observed disrupted group tended to experience correlated recovery more frequently. As the number of addresses in the observed disrupted group increases, do the number of addresses that recover in a correlated manner also increase?

Table 5.3 answers this question. The -2 and -1 columns show events where there is insufficient data from the Thunderping dataset to determine recovery; -2 shows events where none of the addresses in the observed disrupted group responded to Thunderping's pings after the disruption and -1 shows events where only one of the addresses responded to Thunderping's pings after the disruption. The rest of the columns show how many events recovered in a correlated manner. We observe that for the majority of events, irrespective of the number of addresses in the observed disrupted group, more than 50% recover together.



(a) **(b)**

Figure 5.7: (a) The distribution of durations of dependent dropouts for all addresses that recovered in a correlated manner. 60% of addresses recovered in less than an hour. (b) For dependent dropout events where at least two addresses recovered, this shows the number of addresses that recovered on the x-axis and the corresponding recovery duration for the event on the y-axis. Dependent dropout events vary in their duration irrespective of the number of affected addresses.

Recovery times are often shorter than an hour

Next, we turn our attention to the time it takes dependent disruptions to recover. Figure 5.7(a) shows that 60% of recovered addresses recovered in less than an hour. Our technique is able to identify this, because we operate at the precision of the 11-minute time-bins from standard outage detection datasets. Conversely, recent work that finds disruptions spanning an entire calendar hour [12] would miss these disruptions.

Next, we examine whether short recovery durations can be attributable to small disruption events: that is, do the recoveries appear quick because only a

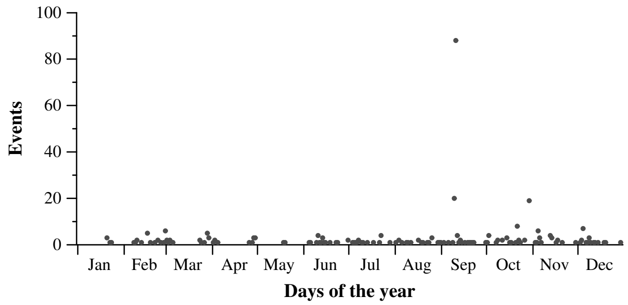


Figure 5.8: Multi-ISP dependent disruption events over time: several ISPs in the same state have simultaneous disruption events on 333 occasions. Here, we show how many events occurred on each day of the year in 2017. Days with many multi-ISP events often correlate with days with large known power outages.

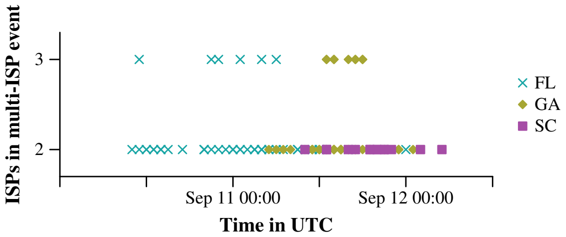


Figure 5.9: Multi-ISP dependent disruption events during Hurricane Irma in Florida (FL), Georgia (GA), and South Carolina (SC). Of 111 events during this time, 15 affected 3 ISPs simultaneously and 96 affected 2.

couple hosts were disrupted? Figure 5.7(b) shows that the answer is no: Even dependent disruptions with hundreds of addresses that recovered together often last less than an hour.

5.4.3 Dependent disruptions can be multi-ISP

Dependent disruption events can also span multiple ISPs within a single state: these events indicate a fault of infrastructure shared by the ISP or their customers.

Here, we broaden our analysis to examine whether the dependent disruption events we detected are correlated across multiple ISPs within the same state.

We observe 333 instances where multiple ISPs in the same state had simultaneous dependent disruption events, and we are able to confirm that many occurred on days when the media reported large power outages in those areas. Figure 5.8 shows days in 2017 when multi-ISP dependent disruption events occurred. Of the 333 instances, 88 (26%) occurred on a single day during Hurricane Irma (Sep 11). Figure 5.9 shows multi-ISP events during Hurricane Irma by state and by the number of individual ISPs affected during each multi-ISP event. We observe 20 multi-ISP events in Florida on Sep 10, when Irma made landfall [105]. As Irma moves northwards, we see multi-ISP events in Georgia and South Carolina as well. Other days with many such events include Oct 30 with 19 events across six states in the Northeastern U.S. (Maine, New Hampshire, Vermont, Connecticut, Massachusetts, Rhode Island); there were recorded power outages during this time as a result of a severe storm [106–108]. On Oct 22, there were 4 multi-ISP events in Oklahoma and 2 in Arkansas; there are corresponding reports of power outages during these times as well [109].

5.4.4 Dependent disruptions may not disrupt entire /24s

Here, we examine if the dependent disruption events that we detected disrupt entire /24 address blocks. If so, they would likely be detected by prior work that looks for outages at these granularities [11, 12]. If there continue to be responding addresses within a /24 with a disrupted address, however, prior work may miss the disruption.

To analyze how dependent disruptions affect /24 address blocks, we find all addresses in the observed disrupted group for a dependent disruption event and group them by /24s. As a running example in this section, consider a dependent disruption event comprising 3 addresses in 1.2.3.0/24, 5 addresses in 2.3.4.0/24, and 2 addresses in 4.5.6.0/24. We call these the *observed disrupted* /24s. For each of these /24s, we also find how many addresses were pinged by Thunderping that were responding to pings *before* the dependent disruption and that continued to respond for at least 30 minutes *after* the time-bin where the dependent disruption occurred. We term these addresses the responsive addresses in a /24 since these addresses were not affected by the disruption.

Our goal is to find how many /24s exist where at least one address was an actual address in a dependent disruption but there were other addresses which continued to be responsive.

First, we check how many of the 20,831 disruption *events* observed at least one responsive address in *all* of the observed disrupted /24s. 12,825 (61%) have at least one responsive address in all of the observed disrupted /24s. For each such event, even if some of the disrupted /24s have addresses that failed independently, since all disrupted /24s continue to have at least one responsive address, prior work may miss detecting this event.

Next, we investigate the subset of observed disrupted /24s where there were at least D_{min} failures within the /24 itself. Since the entire state-ASN ag-

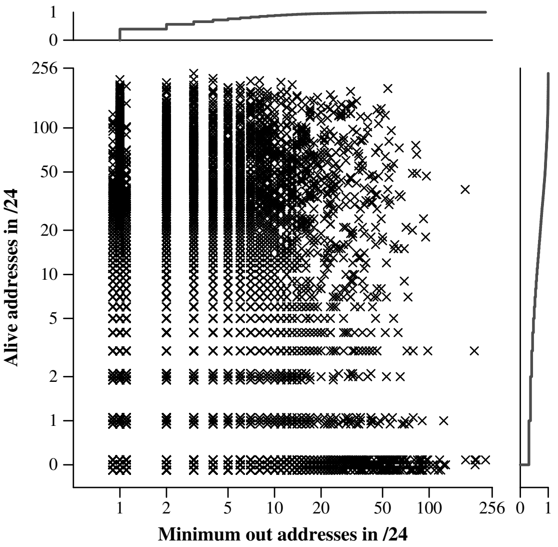
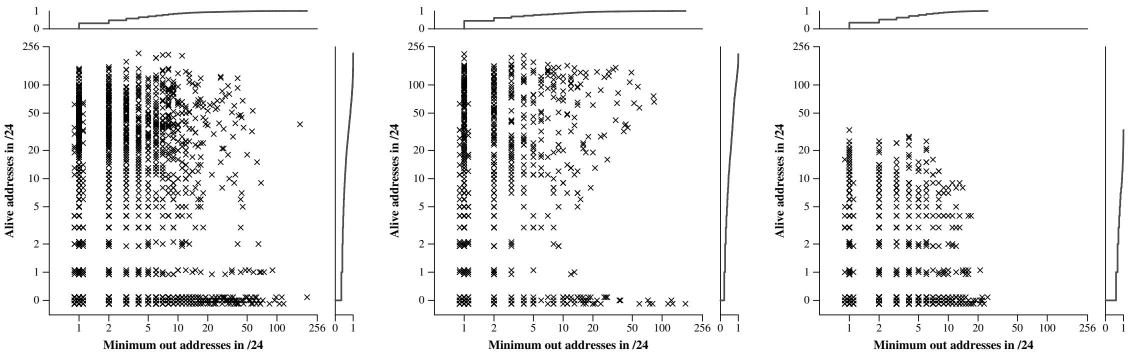


Figure 5.10: Minimum actual disrupted addresses in a /24 vs. responsive addresses in a /24, for all /24s with at least D_{min} address that were disrupted during a detected dependent disruption event.

gregate only required D_{min} failures, when D_{min} or more addresses are disrupted within a single /24, the /24 has at least one actual disrupted addresses. We obtain the minimum bound on the number of actual disrupted addresses in a /24 by subtracting $D_{min} - 1$ from the observed disrupted addresses in that /24. Suppose the D_{min} for the example dependent disruption event above was 3. We would obtain a minimum bound of at least 1 actual disrupted address in 1.2.3.0/24. In 2.3.4.0/24, the lower bound is 3. In 4.5.6.0/24, the lower bound is 0 and we are unable to determine if the addresses in this /24 had a dependent disruption. Of 92,777 /24s with observed disrupted /24s (across all dependent disruption events), we find that 14,702 (16%) have at least D_{min} disrupted addresses. Each of these is a point in Figure 5.10.



(a) Comcast **(b)** Qwest **(c)** Viasat

Figure 5.11: For Comcast, Qwest, and Viasat: Minimum actual disrupted addresses in a /24 vs. responsive addresses in a /24, for all /24s with at least D_{min} address that were disrupted during a detected dependent disruption event. All ISPs have /24s with actual disrupted addresses where there continued to be responsive addresses throughout the disruption.

We find that many disrupted /24s with actual disrupted addresses have other addresses that continued to be responsive. 10,164 (69%) /24s had at least one responsive address, 9327 (63%) had at least two responsive addresses, and 6,096 (41%) had at least 10 responsive addresses. 1,691 /24s had at least 10 actual disrupted addresses; of those, 550 (33%) had at least 10 responsive addresses.

Next, we investigated if the responsiveness of other addresses in /24s with actual disrupted address would vary across ISPs. Figure 5.11 shows per-ISP behavior. We see that all ISPs have /24s with actual disrupted addresses where there continued to be responsive addresses throughout the disruption.

5.5 Conclusion

In this chapter, I showed how to detect dependent residential disruption events using individual address disruptions. Using the binomial test, I detected events where multiple addresses that are related to each other by geography and ISP fail simultaneously such that the failures are unlikely to have occurred independently. The technique is capable of detecting large known disruption events, such as power outages during times of severe thunderstorms, but importantly, can also detect much smaller events. By analyzing these events, I demonstrated that prior techniques which detect dependent disruptions affecting a substantial number of addresses in BGP prefixes or /24 address blocks can miss observing these events. These results motivate finding individual address outages for measuring residential Internet reliability.

Chapter 6: Analyzing weather's effect on Internet Reliability

One aspect of measuring Internet reliability is to determine if the occurrence of certain events adversely affects Internet connectivity. Consider the occurrence of adverse weather conditions for instance: prior work has shown that Internet outages occur more frequently during times of precipitation [5]. However, this work was preliminary in nature and was performed over a short duration (three months).

In this section, I discuss a technique to quantify the effect of external factors, such as the occurrence of various weather conditions, upon Internet connectivity of residential addresses using measurements from the Thunderping probing system [5]. The technique mitigates false outages due to dynamic addressing and user behavior. First, I verify that weather conditions do not positively correlate with peak diurnal failure periods, where dynamic addressing or false outages due to user behavior are common. Next, I quantify the absolute increase in the number of outages observed during weather, when compared to non-weather periods—the outage inflation—for several types of weather including times with precipitation and times with extreme temperatures and high winds. By studying outage inflation of various link types and geographic regions across weather conditions, I am able to identify networks that are vulnerable to weather.

6.1 Introduction

Wather-related damage to vital infrastructure can lead to significant economic harm. Yet, little is known about the economic impact of weather-induced outages on the most pervasive infrastructure that people use to access the Internet: residential last-mile links. For massive last-mile outages, telcos are required by U.S. policy [110] to report the outage to the FCC. However, the minimum reporting threshold is high: the outage must be at least 30 minutes in duration, and it must have affected tens of thousands of customers [110]. Researchers have also studied widespread link failures in the Internet, like undersea cable cuts [111, 112], natural disasters [113], and backbone router failures [114].

In practice, most weather events are much more localized and not severe enough to generate such a large outage. For decades, this everyday weather has been known to lead to to smaller scale outages of telecom infrastructure. For example, early telephone and cable television engineering documents describe how to avoid moisture in wires because it impedes signal propagation [115, 116]. Also, rain attenuates satellite signals above 10 GHz [117]. Finally, point-to-point wireless links can experience multipath fading due to objects moving in the wind [118]. In short, residential links are vulnerable to everyday weather because residential equipment and wiring are often installed outdoors: wind can blow trees onto overhead wires, heat can cause equipment to fail, and rain can seep into underground equipment cabinets.

Surprisingly, for these everyday weather conditions, there are no public statistics on the frequency or magnitude of the outages they induce (directly and indirectly). This could be a problem for Internet-based companies because they do not know how many customers they are losing to nature, and for regulators because they do not know how significant the problem is, and which conditions and geographic areas deserve their attention. In this work we resolve this issue: we provide the first comprehensive study that identifies the correlation between everyday weather and residential Internet last-mile outages. Specifically, we quantify the absolute increase in the number of outages observed during weather, when compared to non-weather periods.

Quantifying the relationship between occurrences of weather and an increase in outages cannot be answered with a short term study. The data set needs to be longitudinal because weather is *seasonal*—certain weather conditions only happen at certain times of year—and because some weather events are *rare* enough that providers in a specific location may not be adequately prepared. Targeted probing is needed because weather is *localized*: at any time only specific geographic locations are exposed to weather conditions. Broad observation of outages of several links will capture correlated outages of several hosts, such as the work by Heidemann et al. [11, 119], but it will not reveal failures of individual links as may be the case for weather. Although some systems can obtain detailed measurements at residential gateways [120, 121], the limited deployment of these measurement systems make them inadequate for studying the scale needed to observe many different weather conditions, multiple times, in different geographic areas. Therefore, we performed a seven year longitudinal study with targeted measurements of residential links surviving weather events.

In 2011, we introduced a measurement system for this task called ThunderPing [122]. For the past seven years ThunderPing has been following forecasts of weather in the U.S. and pinging a sample of 100 hosts from each last-mile provider in the area for six hours before, during, and six hours after the forecasted weather event. The focus of our initial paper on ThunderPing was its probing methodology, but it also included a preliminary study that looked at 66 days of data. Given how limited the data set was, we were unable to draw statistically significant conclusions and we saw only one season, summer, of one year. We also did not have enough data to explore variations in effect of weather by geography, nor could we explore if the likelihood of failure varies with continuous weather conditions (e.g., wind speed),

In this paper, the time totaled across all responsive links exposed to different weather events is in the **centuries**. For example, we have observed a total of *100 centuries* of DSL links exposed to cold weather. This large data set enabled us to address all of these significant limitations of our prior preliminary study.

There is a challenge with quantifying how weather correlates with outages: outages are relatively uncommon events, and thus every outage is a significant event. This is compounded by the fact that we wish to analyze subsets of our data to focus on, say, particular link types or locations. With so few outages observed compared to the time that links are responsive, it is difficult to determine if different weather causes a statistically significant increase in outages. To address this issue, we borrow statistical tools from epidemiology that enable us to reason about the *inflation* in dropout probability, and to establish statistical significance to our results, even though failures happen at relatively low rates. We detail this approach in Section 6.3.1, as we believe it to be of general use to the community.

Another challenge is this metric could be artificially inflated by weather conditions coinciding with daily network state changes such as maintenance or renumbering [21]. We verify that weather does not appear to be positively correlated with peak diurnal failure periods.

Observations and Contributions We present a dataset spanning seven years, all weather conditions, and 76 billion responsive pings to 8.7 million hosts throughout the U.S. We apply techniques from epidemiology to attribute statistically significant rates of dropout to individual weather conditions. Our key findings span four broad areas of analysis:

- **Link type variations** (§6.4.1): Different link types experience weather in highly varying ways. For instance, compared to wired link types (cable, DSL, fiber), wireless link types (WISPs and satellite) experience greater increases in dropout rates during rainy conditions and high temperatures, but often *decreases* in dropout rates in snow and cold temperatures.
- **Geographic variations** (§6.4.2): Different geographic regions can be affected to varying degrees. For instance, Midwestern U.S. states are more prone to

failures in thunderstorms and rain than coastal states. Southern states are more prone to failures in snow than other states.

- **Continuous variable analysis** (§6.4.3): Most link types have highly nonlinear dropout rates with respect to changes in temperature, wind speed, and precipitation. For temperature, dropout rates are typically non-monotonic; satellite links drop out more in moderate temperatures than low or high temperatures.

Our findings have ramifications on how network outage detection and analysis should be performed; limiting measurements to any particular geographic region, link type, or time of year can introduce statistically significant bias. We believe our results also have implications for network administrators and policymakers; an increased use of satellite links in the Midwestern U.S. has resulted in those states' increased dropout rates in rainy weather. We will be making all of our data and code publicly available.

6.2 Data Set

This section describes the data we collected and its initial processing. We start with a definition of the partially interpreted data we seek: "dropouts," where an address fails to respond in the context of otherwise "responsive hours" of an address. Dropouts and *disruptions* from Chapter 5 are synonymous. We next briefly review the ThunderPing data probing system and present brief statistics about the raw active probing data. Then, we review the weather data, particularly how and where it is collected and how we handle hurricanes. We conclude by describing the benefits (and limitations) of this data for our study of weather-related effects.

6.2.1 Dropouts, Defined

A dropout happens when the address attached to a residential link transitions from being responsive to pings from multiple vantage points, to being unresponsive from all of the vantage points. Specifically, we define a residential link "dropout" as an hour when at least three vantage points pinging a host and receiving replies suddenly experience 11 minutes (an entire probing interval) where they do not receive a reply before a five second timeout. This dropout occurs within a "responsive address hour," a continuous observation of an IP address in known weather conditions. A responsive hour may or may not include a dropout, and the ratio of dropouts to responsive hours is a measure of outage likelihood. Responsive hours add: two addresses both observed in the same hour or one address observed for two hours in the same conditions are equivalent.

Our selection of three vantage points is based on prior work's selection of three vantage points to observe outages [11]. Our selection of a five second timeout for ping responses is based on our prior work that observed that most ping replies to residential hosts are received within five seconds [19]. Our selection of one hour as the time period for a dropout is based on the fact that the weather data we collected consists of hourly reports. Considering at most one dropout per probed address per hour will diminish the number of observed dropouts from individual links, if they should alternate between responsive and unresponsive states: there can be at most one per hour, not five (due to the 11 minute probing interval).

Observing a dropout is a sign that a residential link may (but may not) have experienced an outage: *Dropouts are a superset of outages.* Dropouts can also occur if the device re-attaches to the network with a new address after only momentary disconnection, typically through re-association of a PPP session for a DSL modem, but potentially through administrative renumbering of prefixes. For our purposes, we expect these events to occur independent of weather, such that the two events can be studied separately. We confirm that dropouts during typical maintenance intervals are unlikely to correlate with weather in Section 6.3.2.

In short, by observing dropouts, we will be able to observe how residential links behave during weather, at the scale necessary to make quantitative conclusions about weather's effect on residential links in the U.S.

6.2.2 Dropouts, collected

We briefly summarize the methodology of "ThunderPing": our probing system that has been running for seven years. More details about ThunderPing can be found in our preliminary work in IMC 2011 [5].

The ThunderPing probing methodology is as follows: For every forecast of severe weather provided by the US National Weather Service, ThunderPing pings a sample of 100 residential hosts from each provider in the affected region. The affected region is specified by FIPS code, which roughly corresponds to counties in the U.S. The probing starts up to six hours before the forecast event, continues during the event, and terminates six hours after the event, regardless of whether the weather materializes.

The residential hosts ThunderPing pings during each weather event are selected from a master list of residential hosts classified by provider (reverse DNS name) and geographic location (FIPS code). We classify link type by provider, when the provider implies a well-defined link type; (typically rural) providers that use a variety of media types to provide connectivity are included under "All" link types with the rest, but are not classified further. We determine location using a MaxMind database from the same year for choosing which addresses to probe, but from the same month for analysis. Although there are errors in both classifications, a location error would be expected to cause an underestimate of the effect of weather by placing a host not in the forecast region falsely into the area of weather effect.

ThunderPing sends pings to each of these hosts from up to 10 geographically dispersed PlanetLab vantage points every 11 minutes. This interval is due to [119]. When a PlanetLab node fails, we replace it, but if the number of working vantage points drops below three, we discard observations at that time as untrustworthy. When there are at least three, we require that all active vantage points do not have a response in order to label the event as a dropout.

ThunderPing retransmits failed ICMP requests: when a vantage point sees a lack of ping response it retries that ping with an exponential backoff up to 10 times within the 11 minute probing interval. Therefore, a dropout will typically require at least 30 failed ICMP requests.

ThunderPing has been running for seven years, and has collected 76 billion responsive pings to 8.7 million residential addresses.

6.2.3 Weather, classified

To quantify the effect of weather on dropouts, we needed to determine what weather residential links were exposed to when a dropout did or did not occur.

The US National Weather Service (NWS) operates a network of 900 automated "ASOS" weather stations. These weather stations are typically located at airports. The NWS weather stations record hourly observations of 24 weather variables in METAR format and make those available [123].

There are two types of weather information: categories that account for the common precipitation types (e.g., thunderstorm, hail, snow) and continuous variables (e.g., wind speed, precipitation quantity).

We annotate each responsive address hour for an address with the corresponding weather information associated with the geographically closest weather station to that address. Doing so allows us to find the number of responsive hours and dropout address hours in specific weather conditions.

Hurricanes are special Severe events are among the most important failure events for us to study how the Internet is affected, as the Internet is increasingly relied on as the primary mode of communication in an emergency [4]. However, severe events have the potential to overwhelm the typical and obscure interesting observations.

The following hurricanes made US landfall during our measurement: Irene (Aug 26–30, 2011), Isaac (Aug 25–31, 2012), Sandy (Oct 28–Nov 1, 2012), Arthur (Jul 3–5, 2014), Hermine (Sept 1–3, 2016), Matthew (Oct 6–9, 2016), Harvey (Oct 6–9, 2017), Irma (Sept 9–13, 2017), and Nate (Oct 7–10, 2017) [124]. Hurricanes manifest as a combination of weather features and are so pronounced that their contribution to thunderstorm or rain outages would be disproportionate.[1] We thus omit them from categorical weather classification (e.g., Figure 6.2). However, we consider data from Hurricane events when studying continuous variables (inches of rain and wind speed, for example, where these extremes are clearly distinguishable). Collectively, these hurricane times account for less than 3% of responsive address hours and 4% of dropout hours.

6.2.4 Data, Summarized

This data set comprises observations from January 2011 to December 2017, though only 1467 days included sufficiently many operating vantage points to classify a responsive address hour.

We show per-ISP highlights in Table 6.1. We observe major providers such as Comcast, Qwest, and ViaSat in all fifty states (and DC). Of the 1.77 Billion

[1] It is disappointing to realize the irony that the most significant weather events are also the least surprising.

				Dropout	Responsive
	IPs	Airports	States	hours	hours
Cable					
Comcast	2,430,104	476	51	532,493	249,562,477
Charter	654,270	418	47	279,950	95,627,261
Suddenlink	195,398	156	26	158,159	34,684,550
Cox	166,596	270	47	61,318	24,659,573
DSL					
Qwest	592,220	710	51	873,042	99,037,723
Centurylink	342,556	237	33	445,085	83,101,301
Verizon DSL	312,344	201	29	169,078	35,133,098
Megapath	147,860	351	43	206,569	65,436,394
Fiber					
Verizon Fios	415,481	154	23	45,982	48,296,147
GVTC	17,758	7	1	9,113	2,023,618
Dickey	5,229	6	3	6,482	1,114,057
WISP					
RISE Bdbd.	57,021	87	22	40,932	12,442,717
Skyriver	4,187	29	6	5,364	2,032,975
Watch Comm.	4,738	11	2	14,980	1,411,321
Satellite					
ViaSat	161,592	763	51	815,258	29,364,585
SageNet	1,352	65	30	3,762	555,288
All	8,674,043	844	51	9,826,096	1,770,774,634

Table 6.1: Summary of data set for large ISPs classified by link type. "All" comprises data from ISPs not included in this sample. (For this table, we count D.C. as a state.)

responsive address hours from Table 6.1, 139M (8%) were hours where responsive addresses experienced rain, 66M (4%) snow, and 19M (1%) thunderstorm.

Contrasted to our preliminary study [5], this covers nearly 22 times the duration (compared to 66 days), and includes roughly 60 times as many dropout events (likely because those days were in spring and early summer).[2]

6.2.5 Why this data?

Others have studied outages and collected broad IP responsiveness data. Here we describe the benefits of our data, addressing its limitations in Section 6.2.6.

Our data provides a view on outages of individual addresses, including isolated outages of "customer premises" equipment or singly-connected links that are most exposed. We rely on statistics to identify a significant change in likelihood of failure, rather than rely upon large outages of infrastructure common to a larger aggregate prefix to signify significance. Every residential link is wired with its own infrastructure: every residence can have different equipment installed in different ways and has its own resident network administrator. As a concrete example, we expect to observe the effect of water infiltration in the network interface device (the demarcation point connecting premise phone wiring to the provider). (We discuss the flip side of this coin below.)

[2] In the public reviews of the IMC 2011 paper, all of the reviewers stated that they wished the dataset was more comprehensive so conclusions could be made about the effects of weather on residential links.

Our data is of a scale large enough to compare link types, providers, geography, and across time. Seven years of data make it feasible to observe multiple instances of both severe and common weather events. Rare events include a fair number of tornadoes and virtually unique events such as snow in Louisiana. Many observations of similar weather increase the confidence in our dropout probability estimates, making it possible to split the data and identify the differences between, for example, heavy and light rain on wireless ISPs in Kentucky. The sampling approach—providing data for each provider in an area—ensures that even less-used network links and providers are well-represented, permitting a comparison with satellites and wireless ISPs that might be poorly represented in end host measurement probes [16, 120, 121] or when using provider-specific data [10, 125, 126].

Our data includes data from times not subject to interesting weather: the method probes before and after forecast weather alerts. "Typical" weather occurs particularly when the forecast does not materialize or the forecast is for a long-term event (e.g., summer fire warnings). With these measurements, we can establish a baseline for the rate of dropouts in common weather conditions. Probing after the weather also permits measuring recovery time as we wait for previously responsive addresses to return.

Our data is not sensitive to link failures elsewhere in the Internet or to PlanetLab vantage point failures. Restated, with multiple vantage points, catastrophic Internet link outages, such as the fiber cuts during the "Baltimore tunnel fire" in 2001 [127] will only be considered as an outage if all vantage points are unable to communicate with the host over the residential link. As described above, without three active vantage points, we make no decision about address responsiveness.

6.2.6 Dataset Limitations

The essence of ThunderPing is to selectively probe only when there is a weather alert forecast for an area, which biases the data toward time periods where there is some atypical weather present. Obviously, regions that experience temperate weather are unlikely to be represented, and we thus do not attempt to quantify what fraction of all residential network outages are caused by weather. More subtly, during the interval around forecast severe weather, the weather conditions may not be ideal: our estimate of the background dropout rate is likely inflated by proximity to potentially severe weather, thus causing us to underestimate the quantitative effect of that weather.

Our approach relies upon active probing to gain breadth across hundreds of providers, but there are limits to this breadth: providers may administratively filter ICMP requests and home routers may decline to respond. We assume that providers and end hosts that filter are no more or less vulnerable to weather and that these features do not affect our conclusions.

Our data set does not identify the cause of an individual dropout. Our analysis seeks to correlate observations of dropouts with weather events under the expectation that a change in probability of outage is related to the weather. Should a user turn off equipment nightly, this is independent of weather and will not not be a factor; should a user unplug equipment when lightning is nearby, such would contribute to the probability of dropouts in thunderstorm. Residential Internet infrastructure is also explicitly reliant on residential electrical power, and we do not isolate power failures. We expect network service outages to be more common than power outages, for power outages to occur only in the most severe of weather conditions, and for power outages not to correlate with link type.

Finally, AT&T, one of the largest DSL and fiber providers in the US does not assign reverse DNS names to their residential customers. As such, they were not included in our master list of residential links that we probe with ThunderPing.

6.3 Quantifying weather dropouts

In this section we describe our methodology for quantifying how much weather correlates with a change in the probability of residential link dropouts. The goal is to find a metric that can measure how the likelihood of a dropout *increases* (or not) during weather.

The challenge in computing this metric is, dropouts are uncommon. This makes it difficult to demonstrate that there is a statistically significant increase in dropouts during weather. Even if there is an increase, it may simply be due to the fact that certain types of weather tend to occur during time periods that are commonly used for network maintenance or IP renumbering.

By leveraging statistics developed for epidemiology, we overcome the first challenge and find statistical significance. By carefully inspecting our data set, we verify that no type of weather that we study correlates with diurnal variations dropout probability.

We will now describe our metric, inflated probability of dropout, then we will verify that conclusions we draw from this metric will not be skewed due to time-correlated network state changes.

6.3.1 Metric: Inflated probability of dropout

Challenge: Dropouts are uncommon

Correlating dropouts with weather is challenging to do with statistical significance because *dropouts are rare events*[3]. On average, we observed a link dropping out only once every $2 - 30$ days that we were actively pinging and receiving responses from the host residential link, depending on the link type. The inverse of the average dropout rate per link type—including the average across *all* link types—is as follows:

Link type:	Fiber	Cable	WISP	DSL	Sat	*All*
Days b/w dropouts:	30	16	8	9	2	*8*

Given how rarely dropouts occur, we now describe a metric that accounts for this phenomenon, and even provides a way to determine statistical signifi-

[3] ...and they should be. If dropouts were common, residential links would be unusable.

cance so that we can ensure that our analysis does not involve too much slicing and dicing of the few dropouts that occur.

Our Approach: Hazard Rate

Fortunately, there is a well-established set of techniques from the field of epidemiology that permit statistical significance over rare events. Epidemiology—the study of the occurrence and determinants of disease—faces similar challenges when analyzing mortality: deaths ("failures") are rare, and subjects ("links") can be exposed to their disease ("weather condition") for different amounts of time until the time of death ("dropout"). Here, we describe the techniques we borrow from biostatistics [102] to address these concerns. Throughout our study, we will consider different groups of "subjects": link types, geographic regions, and combinations thereof.

Like in epidemiological studies, we focus on estimating the *hazard rate* (sometimes referred to as the instantaneous death rate). In essence, what a hazard rate gives us is the expected number of deaths per unit time. More concretely, for a given hazard rate λ, the probability of death over a short duration of time t is $\lambda \cdot t$.

The first challenge in estimating hazard rates is that different subjects may be observed over different periods of time: in our study, hosts that remain responsive can naturally be observed for longer periods of time than those that drop out. Throughout an observation period, we track the amount of time O_i that we observe each host $i = 1, \ldots, n$, and we also count the total number of dropouts, F.

An *unbiased* estimate of the hazard rate $\hat{\lambda}$ can be obtained as follows [102, Chapter 15.4]:

$$\hat{\lambda} = \frac{F}{\sum_{i=1}^{n} O_i} \tag{6.1}$$

We exclude any bin of data if it does not have enough samples to permit computing confidence intervals. We adhere to the following rule of thumb [102, Chapter 6]: we accept a bin with n samples and estimated hazard rate $\hat{\lambda}$ only if

$$n \geq 20 \quad \text{and} \quad n\hat{\lambda}(1-\hat{\lambda}) \geq 10 \tag{6.2}$$

When these conditions hold, we can calculate 95% confidence intervals over the estimated hazard rate as follows [102, Chapter 6.3]:

$$\hat{\lambda} \pm 1.96 \cdot \sqrt{\frac{\hat{\lambda}(1-\hat{\lambda})}{n}} \tag{6.3}$$

The above calculations yield the hazard rate along with its confidence intervals; what remains is to *compare* two hazard rates, for instance, the overall hazard rate for a given link type and the hazard rate for that link type specifically in the presence of snow. Two estimated hazard rates $\widehat{\lambda_1}$ and $\widehat{\lambda_2}$ can be compared by simply subtracting them [102]. Fortunately, with sufficiently many samples, the confidence intervals over the difference of two hazard rates is given by the addition of the confidence intervals over the original hazard rates.[4]

[4] This follows from the fact that var$(\lambda_1 - \lambda_2)$ is approximately var$(\lambda_1) +$ var(λ_2) when Eq. (6.2) holds.

To summarize: in the results throughout this paper, we compute hazard rates using Eq. (6.1), discard any bins that do not satisfy Eq. (6.2), and compute confidence intervals using Eq. (6.3). When presenting our results, we multiply the hazard rate by a short time interval, one hour, to estimate the hourly probability of a dropout.

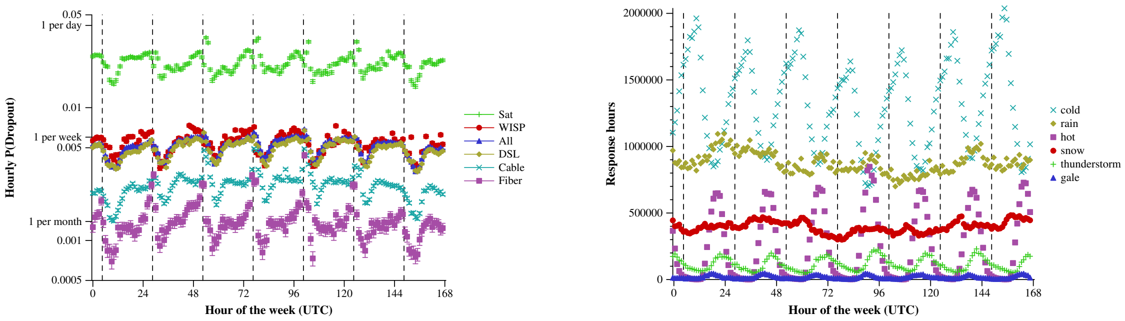
6.3.2 Verifying the metric will not be skewed by common dropouts correlating with weather

Challenge: Dropout probability in weather may be inflated due to diurnal events

Observing an increase in dropout probability during weather periods compared to non-weather periods may be skewed by common network state changes that tend to occur during certain types of weather events. This is a significant problem because it is more likely that a dropout would be caused by everyday changes in network state such as nightly maintenance periods, IP renumbering, and customers powering off their links at night, rather than weather-induced outages.

Our Approach: Verify that weather events do not positively correlate with common dropout periods

The first question we must answer is: Are there any hours of the week that have a significantly higher probability of dropout than other hours of the week? To



(a) Dropout probability has significant diurnal variation.

(b) Different weather conditions are prominent at different times.

Figure 6.1: Weather does not occur most often during hours of the week when there are an inflated number of dropouts.

answer this question, we evaluate the probability of dropouts in each hour of the week in the following manner: for each hour of the week, we counted the number of dropouts (recall that dropouts only occur at most once per hour per link) across all links observed during that hour, then we divided that by the number of hours which the link was responsive. We did this for each link type separately, as some link types may be more likely to be renumbered. For example, in prior work we discovered that European DSL links have their IP addresses reassigned every night at 2:00 AM UTC [21]. Also, some link types may require maintenance more often than others.

The results are shown in Figure 6.1(a). As expected, the hourly probability of dropouts significantly varies in a diurnal pattern over the course of each week. Prior work suggests that ISPs are more likely to perform administrative maintenance during weekday night hours [128, 129]; we speculate that the increased dropout probability during weekday night hours could be due to administrative maintenance. The highest probability of dropouts for every link type rises in the evening and peaks near midnight Eastern Standard Time (indicated with vertical dotted black lines), after which it drops off significantly until the early hours of the morning.

Given that we observe a diurnal variation in hourly likelihood of dropouts, and the fact that weather conditions also have a known diurnal pattern of occurrence [130], the next question we must answer is: Does hourly weather occurrence positively correlate with dropout probability? To answer this question, we count the total number of responsive hours that we observed in each hour of the week for each weather condition.

The results are shown in Figure 6.1(b). As expected, most weather conditions, possibly except for snow, have a diurnal pattern in their occurrence. Fortunately, none of the weather conditions have a positive correlation with the hourly probability of dropouts. There is however a negative correlation with cold weather: the coldest point of the night is also when the lowest hourly probably of dropouts occurred. This negative correlation will not have an effect on the quantified failure rate, as dropouts are less likely to occur during the hours when it is cold than during other hours.

Baseline probability of dropout depends on link type

The investigation into probability of dropout for each link type also provides additional justification for the selection of a metric that is based on the *increase* in failure probability due to weather. The dropout probability significantly different for each link type, with Fiber being the lowest and Satellite being the highest (Figure 6.1(a)). With this metric, the baseline failure rate will be removed from all link types; including the diurnal variations in dropout probability.

6.3.3 Summary

We selected a hazard rate-based metric that enables us to study the statistically significant *increase* in dropout probability during weather events. Then we verified that this metric will not be skewed by nightly spikes in dropout probability because they do not correlate with occurrence of weather conditions.

6.4 Weather Analysis

In this section, we use our collected data to understand how weather conditions affect dropouts.

6.4.1 Relative dropout rates

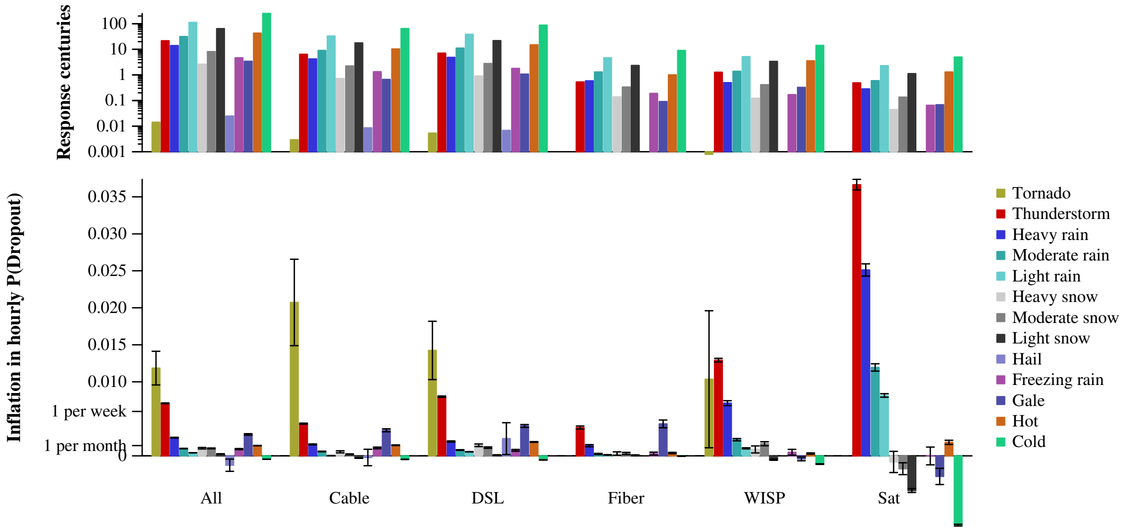


Figure 6.2: The number of response-hours (in centuries) for which we have measured various link types in various weather conditions (top), and the additional ("inflated") probability of dropout experienced in those link- and weather-types (bottom).

First, we analyze the relative rate of dropouts under various link types and weather conditions, after omitting all hurricane periods. We use categorical data from weather records (such as "thunderstorm present"), to assign a single weather condition for each hour. If more than one weather condition occurred in an hour, then we assign the most severe condition to that hour.

The top of Figure 6.2 shows the number of responsive hours for which we measured the various link- and weather-types. Although there is a wide range in their absolute values (note the log-scale of the y-axis), the overall shape of the histograms remains mostly consistent across the different link types. This reflects the fact that, in their deployment throughout the US, different link types are exposed to very similar conditions, with one minor exception: we did not measure any fiber or satellite links during tornadoes.

The bottom of Figure 6.2 shows the difference in the probability of failure rate between the presence of a weather condition and its absence. A value of zero signifies no observed difference with or without a particular weather condition; positive values indicate increased probability of dropout during that weather condition; and negative values indicate *fewer* failures during that weather condition. In the bottom of the figure, we also include confidence intervals on all bars; they are tight on almost all values, but satellite links are noticeably variable, as are tornadoes.

We make three key observations from Figure 6.2. First, there are several weather conditions that exhibit higher dropout probabilities across *all* of the link types we measured. Thunderstorms, heavy rain, moderate rain, and (for the link types that experienced it) tornadoes all yield a statistically significant increase in dropout probabilities.

Second, for each given link type, heavier rates of precipitation (both rain and snow) yield higher probabilities of dropout. We analyze dropout rates as a function of precipitation in Section 6.4.3. Interestingly, the probability of dropouts is greater during thunderstorms than during heavy rain for all link types. Recall that we classify "thunderstorm" and "heavy rain" as mutually exclusive. This indicates that the causes of failures during thunderstorms extend beyond the rainfall, perhaps to increased wind or power outages.

Finally, the dropout probabilities of wired link types (cable, DSL, and fiber) are similar to one another, as are the dropout probabilities of wireless link types (WISP and satellite), but wired and wireless link types are different from one

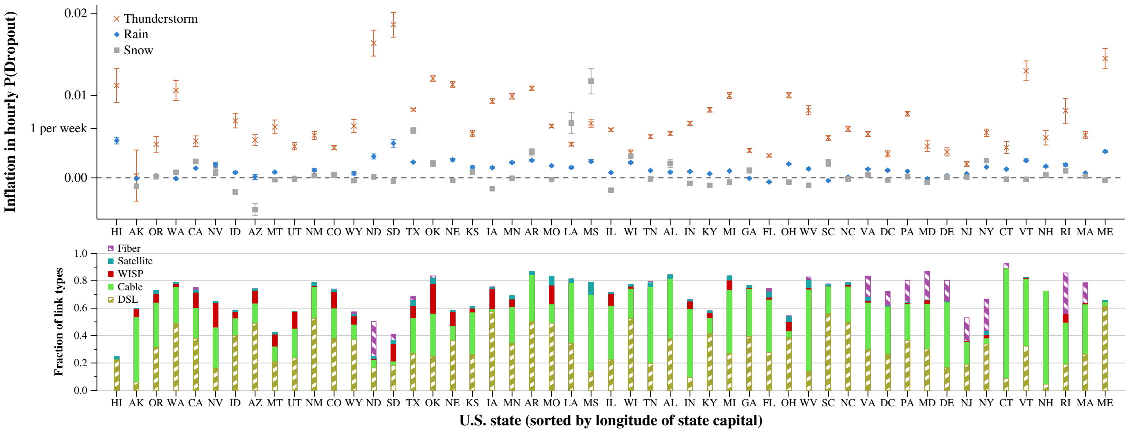
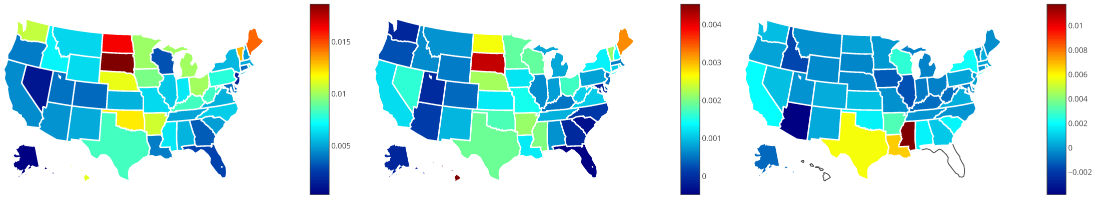


Figure 6.3: Top: Inflation in hourly dropout probability by U.S. state for thunderstorm, rain, and snow (with 95% confidence intervals). Bottom: The fraction of link types by U.S. state (the remaining fraction are of unknown type).

another. For example, light rain and light snow have almost no discernible difference in dropout probabilities for wired links, but light rain exhibits higher dropout probability for wireless links, and light snow sees *lower* probability of dropout. Conversely, gale-force winds have a profound increase in dropout probabilities for wired links, but wireless links are less likely to drop out during them. It is not surprising that strong winds can cause wired links to fail, for instance by knocking down above-ground cables. Although wireless links are not affected in the same way, it is surprising that higher failure rates would not be observed, given that such strong winds could destroy or blow away satellite dishes.

Summary and ramifications The results from Figure 6.2 collectively show that different link types can experience weather in different ways. It is not surprising that different link types would differ in the *magnitude* with which they experi-



(a) Thunderstorm **(b)** Rain **(c)** Snow

Figure 6.4: Inflation in hourly dropout probability by U.S. state for various weather conditions. Large geographic regions can exhibit common behavior; northern states are more prone to failures in thunderstorms, Midwestern states in rain, and southern states in snow. (Note the different scales for each sub-figure.)

ence dropouts; but what we do find surprising is that some weather conditions (especially snow and cold) can differ in whether they increase or *decrease* dropout rates. This has ramifications on network measurement methodology: when performing outage analysis, it is important to account for both link type *and* weather condition.

6.4.2 Geographic variation

Next, we investigate the extent to which different geographic regions experience weather in different ways. Of course, different states experience different *amounts* of weather (for instance, we did not observe a statistically significant amount of snow in Florida). To control for this, we present the inflated probability of hourly dropouts, comparing hours with a particular weather condition (e.g., snow) against all hours without that weather condition. This gives us an apples-to-apples comparison across states, even if they experience weather conditions in varying amounts.

In Figure 6.3, we present the dropout probability inflation across all 50 U.S. states (and DC) for three weather conditions: thunderstorms, rain (excluding hurricanes), and snow. We make two key observations. First, there is a high variation of increased dropout probability across states. For example, during thunderstorms, South Dakota experiences an average increased hourly dropout probability of 0.018 (3.1 additional failures per week), while New Jersey increases by only 0.0038 (2.9 additional failures per *month*). Moreover, as shown by the 95% confidence intervals in the figure, these differences are statistically significant. We believe this to be an important result because it shows the role that geography plays in network outages.

Second, while the raw dropout inflation varies among states, the relative impact of weather types is common across *most* states: the increase in dropouts during thunderstorms tends to be greater than in rain, which in turn tends to be greater than in snow. There are a few notable exceptions. Louisiana and Mississippi have more inflated dropouts in snow than in thunderstorms, and Florida tends to experience similar amounts of failures in rain as it does in thunderstorms. By controlling for geography and the total amount of time spent in weather, this result shows that some weather conditions have more pronounced impact on dropouts.

Below Figure 6.3, we present a breakdown of the classified link types in each state, weighted by responsive hours in probing. The intent of consulting this graph is to determine whether the outliers in the top graph are a direct function of the link types that are prevalent in a state. North Dakota has a substantial and exceptional deployment of Fiber: 50% of the link-type-classified responsive hours are from Fiber addresses. Although our sampling approach is based on finding 100 addresses in each provider in a region, and thus is not meant to sample the distribution of link types used by customers, we note that this is consistent with published reports that "60 percent of the households, including those on farms in far-flung areas, have fiber" [131]. Although there are instances where top and bottom graphs appear related—Vermont (VT) and Maine (ME) show both a high vulnerability to thunderstorms and a relatively large proportion of DSL compared to immediate neighbors CT, NH, MA—it appears that geography is more important than link type at determining the inflation in probability of dropout in precipitation.

Next, we look beyond individual states to see if there are *regional* correlations of dropouts. In Figure 6.4, we show maps with the average inflation in dropout probabilities during thunderstorms, rain (excluding hurricanes), and snow.

During thunderstorms (Figure 6.4(a)) and rain (Figure 6.4(b)) Midwestern states tend to experience greater inflation of dropouts than other regions. (Maine is an outlier; its dropout inflation during thunderstorm and rain is due to an abnormally powerful series of storms in October 2017.) Recall from Figure 6.2 that WISP and satellite links fail more often in thunderstorms and rain than other link types. One possible explanation for higher dropout rates in the Midwest would be that these states have more wireless links. This hypothesis is confirmed in Figure 6.2, which shows that Midwestern states have more satellite links than other states.

During snow (Figure 6.4(c)), we see more pronounced dropout inflation in southern states.[5] Texas, Louisiana, and Mississippi experienced drastically higher probability of dropouts in snow than in the absence of snow. Unlike rain and thunderstorm, this disparity cannot be explained by link type alone, as no link types experience drastically higher dropout rates than others (in fact, Figure 6.2 shows that wireless links tend to experience *fewer* dropouts during snow). Our insight is that snow seems to affect states where snow is less common.

One possible explanation for the regional effects is therefore that regions that are less "familiar" with a particular weather condition may be more heavily affected by it. To evaluate this hypothesis, we plot in Figure 6.5 the hourly dropout rate of each U.S. airport as a function of the number of hours each airport has spent in snow. The results in this figure confirm our hypothesis for snow: the less familiar a location is to snow, the more often it tends to experience dropouts. Areas with very small amounts of snow do not experience large inflation (ostensibly because there is not enough time for it to cause damage). Conversely, areas with snow beyond a threshold are more resilient to snow. A likely reason for this is that regions that are more used to snow tend to invest more in infrastructure to prepare for and mitigate it [132]. We also performed this analysis under thun-

[5] We do not include data for Florida or Hawaii, as we did not observe enough responsive hours of snow to achieve statistical significance.

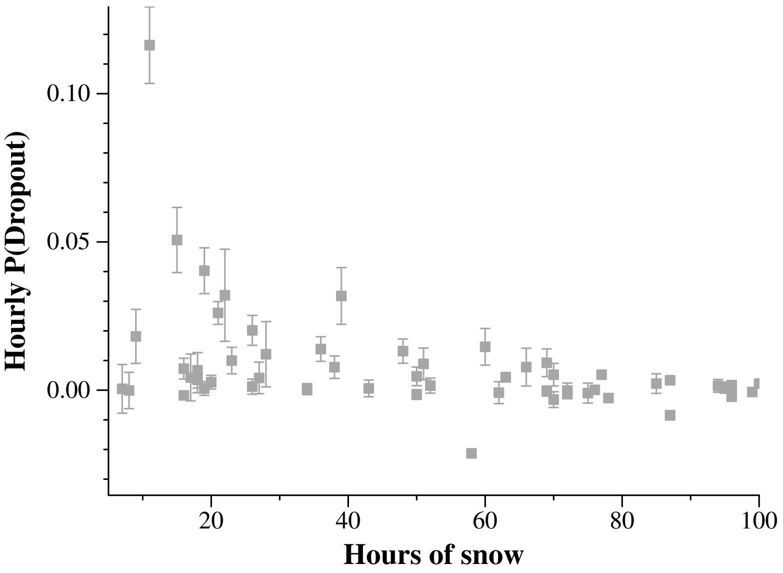


Figure 6.5: Hourly dropout probability of hosts (all link types) as a function of the number of hours the hosts' nearest U.S. airport received snow (truncated to only those with fewer than 100 hours in snow). The less common snow is in a region, the more impact it tends to have.

derstorms and rain (figures not shown), but did not observe as strong an effect. We hypothesize that this is because all of the airports we measured experienced enough thunderstorm and rain to grow accustomed to them.

Summary and ramifications We conclude from these results that different geographic regions can be affected by weather to varying degrees. We attribute this geographic variation to two leading factors: (1) the predominance of some link types over others (e.g., wireless links are more common in the Midwest), and (2) how familiar a region is with a particular weather condition (and thus how prepared for it the region is). Our results have several interesting ramifications on outage analysis. First, when performing outage analysis, it is important to consider a representative set of locations and link types; measuring only, say, cable links would risk overestimating the Midwest's resilience to dropouts. Second, it is important to note the time and weather conditions when outage measurements are taken; collecting measurements only during Spring months[6], when thunderstorms are more common, would risk overestimating dropouts year-round.

6.4.3 Continuous weather variables

Thus far in our analysis, we have considered various *binary* classifications of weather—rain (or not), snow (or not), gale (or not), and so on. Although these classifications are standard (they are included in the weather reports we collect), they risk masking the precise effect that various weather conditions can have. Here, we evaluate dropouts as a function of several *continuous* weather variables: wind speed, precipitation, and temperature.

Figure 6.6 shows the inflation in the hourly dropout probability of various link types as a function of wind speed. Note that not all link types share the same values on the x-axis; we aggregate data in increasing values of x until we reach either an interval of 10 mph or 20 dropout samples (Eq. (6.2)), whichever comes last.

For all link types, we see almost no inflation in dropout probability when wind speed is less than 30 mph. Beyond 30 mph, there is little effect on wireless links (WISP and satellite), but significant increases in dropout probability for wired links: cable, DSL, and fiber. This is reflected in Figure 6.2, which showed

[6] For instance, in the run-up to the IMC deadline.

that wireless links were not as affected by gale winds. Figure 6.6 expands on this by showing that, as wind speed increases, dropout inflation increases at a super-linear rate—between 40 mph and 55 mph winds, Cable links' dropout inflation increases by an hour of magnitude.

In Figure 6.7, we show dropout inflation as a function of temperature. Like with wind speed, we bin along the x-axis in units of 10, or 20 dropout samples, whichever comes second, and include 95% confidence intervals. There are several surprising observations in this figure. First, satellite links are highly sensitive to temperature; at low temperatures, satellite links are far less likely to experience dropouts, but this increases steadily, until at approximately 70° F when satellite links become more likely to fail. Surprisingly, at approximately 80° F, there is an inflection point at which satellite links again become significantly more reliable. We hypothesize that there is a confounding factor: satellite links are less reliable when there is no line-of-sight visibility (e.g., due to fog), and we suspect that higher temperatures result in less fog.

All of the other link types we measured exhibit similar behavior to one another. They have highly variable dropout probabilities at low temperatures; they remain mostly steady until 60° F, then they increase slightly with higher temperatures. Unlike with our other results, WISPs more closely resemble wired links than satellite links; we hypothesize that this, too, is because satellite links are affected by line-of-sight while WISPs and wired links are not.

Finally, in Figure 6.8 we measure various link types' dropout inflation as a function of precipitation in thunderstorms, rain, and snow. All link types exhibit

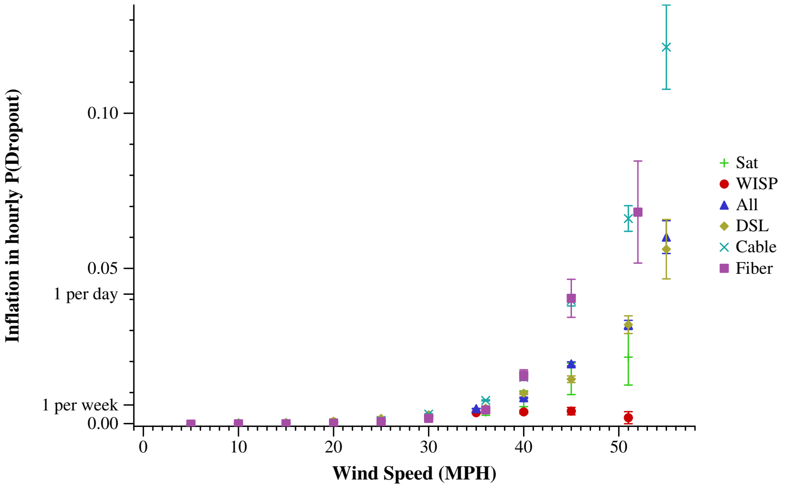


Figure 6.6: Inflation in hourly dropout probability as a function of wind speed across multiple link types. All link types experience greater dropout probabilities, but satellite and WISP links increase the least.

increased dropout inflation with increased precipitation, regardless of the overarching weather condition. However, surprisingly, the magnitude of increase varies significantly across link types. Again, satellite tends to be the most sensitive to change. Other link types are not as consistent across different types of precipitation; WISP links exhibit nearly the same increase in dropouts at high thunderstorm precipitation as satellite, but far less during non-thunderstorm rain. Similarly, DSL links experience a (varying but statistically significant) increase in dropouts during high snow precipitation, but not nearly as much during thunderstorm or rain.

There appears to be an inflection point with snow and rain: prior to 0.1 inches of precipitation in rain or snow, non-satellite links experience little change in their dropout probabilities. After these points, they increase significantly and quickly.

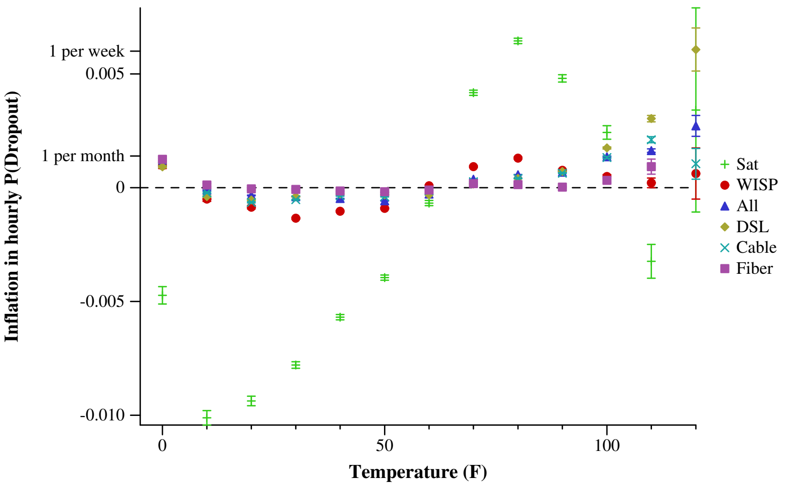


Figure 6.7: Inflation in hourly dropout probability as a function of temperature across multiple link types. All link types exhibit non-monotonic effects, typically increasing at higher and lower temperatures (satellite being a clear exception).

Conversely, most links experience (slight but statistically significant) increases in dropout rates in all levels of precipitation during thunderstorms.

Summary and ramifications Weather conditions are often described with binary categories: rain (or not), snow (or not), and so on. These continuous variable results show that such categories can be overly coarse; the mere presence of rain or snow does not necessarily affect most link types, unless there is more than 0.1 inches of precipitation. Like with our prior results, different link types can exhibit widely varying behaviors, lending further motivation to incorporate link types into future outage analyses.

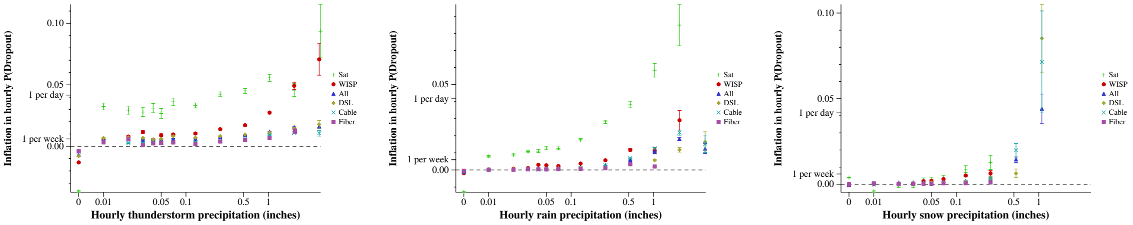


Figure 6.8: Inflation in hourly dropout probability as a function of precipitation during thunderstorm (left), rain (center), and snow (right), across multiple link types. All link types experience higher dropout probabilities with more precipitation, but to widely varying magnitudes. (Note the different ranges of the x-axes.)

6.5 Conclusions

Using a seven year dataset collected by probing residential IP addresses in the U.S., I showed that a variety of weather conditions can inflate the likelihood of Internet dropouts. I quantified this inflation and show that it varies depending upon the type of weather, link type, and geographic location.

Even ignoring times when hurricanes were active, all link types see more failures during thunderstorms—fiber addresses, the most resilient to thunderstorms still observed an additional dropout every 11 days, while satellite addresses, the most susceptible, observed an additional dropout every day. High wind speeds result in a super-linear increase in dropout probability for wired links while higher precipitation results in particularly pronounced increases in dropout probability for wireless links.

The extent to which weather conditions can inflate the probability of dropouts varies considerably with geography. States in the Midwest are susceptible to dropouts during rain while states in the south experience dropouts much more often in the snow: addresses in Mississippi, for example, experience an additional dropout every 4 days.

The reliability analyses in this chapter were performed using dropouts of individual IP addresses. Although dynamic addressing and user behavior also constitute dropouts, the key observation that allowed the use of dropouts for reliability comparisons is that the inflation in dropout rates during the occurrence of severe weather conditions is due to the additional outages that occur. Confounding factors such as dynamic addressing and user behavior do not positively correlate with peak diurnal failure periods; therefore, the increase in dropout rate during a weather condition is equivalent to the increase in outage rate during that condition.

Chapter 7: Future Work

Here, I identify directions for future work in measuring residential reliability using probing-based techniques.

7.1 Tracking devices across IP addresses using IDs on a global scale

In Chapter 4, I showed how to use IDs from complementary datasets to (i) analyze dynamic addressing patterns and (ii) to confirm outages. However, the RIPE Atlas dataset consisted of ten thousand probes, a small fraction of residential users around the world. While the CDN dataset was obtained from an order of magnitude more devices than RIPE Atlas, it could still only offer confirmation for 1% of Thunderping's detected outages.

With more sources of IDs, it may be possible to model the likelihood of address change. Such models can help prevent false inferences about outages and their durations. For ISPs that change periodically and/or synchronously, the model can predict when probe-loss is more likely due to address changes than outages. For ISPs that change addresses upon most outages, the model can inform in which ISPs outage duration detection is particularly error-prone. For other ISPs which change addresses mostly upon longer outages, the model can be used to estimate the likelihood that an inferred outage ended falsely.

Orthogonally, if every outage detected by a probing-based technique could be confirmed through a complementary dataset that provides IDs, false positives due to dynamic addressing can be entirely mitigated. Additionally, the analysis of outage recovery durations for all of these outages will be possible.

The key to tracking IP address(es) assigned to a home router over time is to associate some uniquely identifying feature (the ID) that remains constant across the home router's address changes. Chapter 4 showed two examples of IDs: in the case of RIPE Atlas probes', the probe ID remained unchanged and in the case of the CDN software, the installation ID remained unchanged.

A challenge, however, is to obtain sources of IDs that can scale across the Internet. IP address changes can be tracked over time if there exists some uniquely identifying feature that remains constant across the device's address change. There are several potential datasets which have this property:

7.1.1 Dynamic DNS services

Websites such as dyn.com [133] provide dynamic DNS. Dynamic DNS is a service that allows users with a dynamic IP address to host web services, by providing DNS services that can be easily updated to reflect changes in users' IP addresses. Users of Dynamic DNS Services run a daemon provided by the dynamic DNS

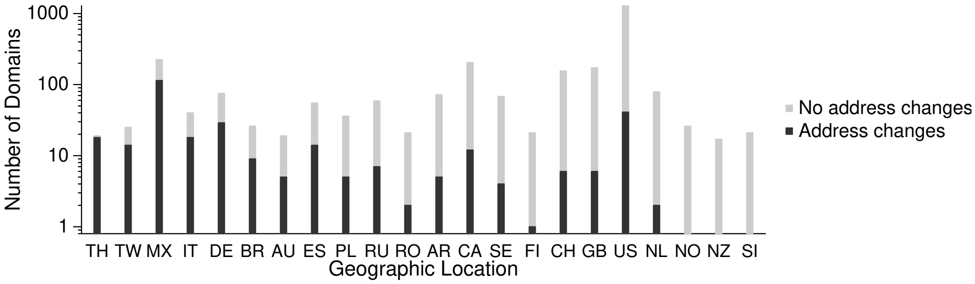


Figure 7.1: IP address renumbering in dynamic DNS domains over a week: Black represents dynamic DNS domains which experienced at least one address change, while grey represents domains whose addresses remained the same. Renumbering behavior appears to be correlated with geographic location.

provider, which is responsible for determining the publicly visible IP address, and updating the A record(s) for the user's domain(s).

IP address changes can be tracked using the domain names registered with dynamic DNS services. Since the domain name of a user maps to her current IP address, we can use the domain name as a fingerprint, and detect changes in IP addresses for each domain name over time, by periodically obtaining the 'A' record associated with each domain name.

Geographic correlation of dynamic behavior As a proof of concept, I report on a preliminary result from this approach: corroborating the geographic relationships in Figure 4.1 while extending to countries not well represented by RIPE. I obtained 3000 dynamic DNS domains from three different dynamic DNS services: 2000 from afraid.org [134], 600 from dyn [133] and 400 from noip.com [135] and fetched the 'A' records from their respective nameservers once every hour.

I collected this data for a week, and then inspected how many of these domains experienced at least one address change during this time. Figure 7.1 shows the number of domains that had at least one address change and the domains that had none. The y-axis is in log-scale. Address changes in Asian and Latin American countries appear more prevalent, with more than a third of all domains in these countries seeing at least one address-change. On the other hand, Northern European countries observe fewer than 6% of their domain names experiencing an address change. Address changes are uncommon in North America: only 3% of domain names in the US and 6% of domain names in Canada observed an address change.

7.1.2 Open DNS resolvers

Since 2010, various studies have reported on the existence of more than 15 million 'open' DNS resolvers on the Internet [69, 136–138]. These DNS resolvers are 'open' because they will resolve a DNS query sent from arbitrary IP addresses on the Internet. Previous studies have found that more than three-quarters of open DNS resolvers are likely to be residential [137, 139]. I identify two potential approaches to fingerprint these open DNS resolvers and track address changes.

DNS caches Open DNS resolvers often cache previous lookups [139]. These caches can be used to fingerprint open DNS resolvers, allowing us to track when their IP addresses change.

Anomalous Open DNS Resolvers Of the 30 million Open DNS Resolvers on the Internet, around 17 million are *anomalous* [68], i.e., instead of sending DNS responses with a source port of 53, they respond with a non-standard source port. Kaizer et al. [68] found that these devices are primarily residential ADSL modems. Not only do these devices use a non-standard source port, DNS requests can be made to these devices in such a way that the source ports are assigned *sequentially*. We can use this sequential assignment of source ports to fingerprint anomalous open DNS resolvers.

7.2 Classifying IP addresses

Probing-based techniques that seek to detect residential Internet outages need a list of addresses classified as residential. More broadly, a classification of the IP address space into residential, enterprise, campus etc., can benefit any system that uses IP addresses as a proxy for measurement, including IP address based host-reputation systems [75, 76]. Recent work has also shown that ISPs are increasingly likely to deploy Carrier Grade NATs (CGNs), where tens of residential Internet connections are multiplexed over a single public IPv4 address [20].

In this dissertation, I relied upon classifications of addresses as residential using reverse DNS based schemes from prior work [5]. Many ISPs include hints about an address's intended use in the reverse DNS entry of that address. Recent research has further improved address classification with reverse DNS names [140]. However, it is not mandatory for ISPs to provide meaningful reverse

DNS names. Some large ISPs, such as AT&T do not provide reverse DNS names for most of their addresses, resulting in their addresses' under-representation in Thunderping data as seen in Chapter 6.

An orthogonal approach to address classification is to use datasets with some uniquely identifying feature (an ID) that can be used to track IP addresses over time. By analyzing how many IDs are associated with an IP address simultaneously and over time, I show in preliminary work that it is possible to infer how the ISP is using the address [141, 142]. An address that is observed with multiple devices over time, though with relatively few devices at any instant, is likely a dynamic residential address. An address that remains associated with a single ID over months is either statically assigned or is a residential address with a linktype that uses DHCP. Addresses associated with many IDs simultaneously could be CGN addresses or university/enterprise proxies.

7.3 Identifying outage causes can help orthogonal reliability analyses

In this dissertation, I covered one possible reliability analysis: examining how challenging conditions like weather affects Internet reliability. Another potential analysis is the head-to-head comparison of one ISP's reliability against another. Such comparisons can aid users in their choice of ISP and can help ISPs gauge their competition.

When comparing reliability across ISPs, the reliability metric should ideally only consider outages that each ISP was responsible for. If a user voluntarily chooses to power off their home Internet equipment, the user has an Internet outage but this outage should not lower the user's ISP's Internet reliability. Similarly, a power outage in an area should contribute towards lowering the reliability of that area, but should not lower the reliability of the ISPs whose addresses were affected.

For conducting comparisons of ISPs, we need to classify outages by detected cause. Chapter 5 showed the potential of using simultaneous outages of related addresses to find addresses that failed due to a common underlying cause. When addresses from multiple ISPs fail together in the same geographic region, the cause is potentially a power outage. When addresses from only a single ISP have been observed to fail, the cause is potentially a network outage. Once outages have been classified by cause, outages in appropriate classes can be used to determine the outage rate per ISP.

Chapter 8: Conclusion

In this dissertation, I described how to measure residential Internet reliability remotely using probing-based techniques. While having the ability to measure broadly, these techniques' outage inferences can be inaccurate. My contributions have improved their accuracy, and have allowed their detected outages to be used in metrics for comparing residential Internet reliability of various ISPs, media types, and geographic regions in different weather conditions. I showed how to detect Internet outages accurately using probing-based techniques by analyzing and mitigating potential scenarios that can cause these techniques to make false inferences about detected outages. In Chapter 3, I investigated how frequently probe responses can be delayed beyond commonly used timeouts by analyzing ping response latencies from IP addresses across the world in a variety of networks. In Chapter 4, I analyzed dynamic addressing patterns in ISPs to find networks where addresses are stable. I also showed how to detect outages in networks where dynamic reassignment is common, using complementary datasets that can provide information on whether a device's IP address has changed. Chapter 5 demonstrated the need for detecting individual address outages when measuring residential reliability. In Chapter 6, I compared the reliability of ISPs, media-types, and geographic regions across several weather conditions.

Bibliography

[1] Inc. Ideal Life. Ideal Life - Remote Patient Monitoring for Home Health Care. http://www.ideallife.com/how-it-works/.

[2] Susanna Spinsante and Ennio Gambi. Remote health monitoring for elderly through interactive television. *BioMedical Engineering Online*, 11(1):54, 2012.

[3] Voipfone. VoIP 999 Emergency Services. http://www.voipfone.co.uk/999_Emergency_Services.php.

[4] Federal Communications Commission. VoIP and 911 Service. https://www.fcc.gov/consumers/guides/voip-and-911-service.

[5] Aaron Schulman and Neil Spring. Pingin' in the rain. In *IMC*, 2011.

[6] Measuring Broadband America. https://www.fcc.gov/general/measuring-broadband-america.

[7] Broadband Measurement Project, Canada. https://crtc.gc.ca/eng/internet/proj.htm.

[8] Broadband in the U.K.: data and research. https://www.ofcom.org.uk/research-and-data/telecoms-research/broadband-research.

[9] Measuring Broadband Australia. https://www.accc.gov.au/consumers/internet-phone/monitoring-broadband-performance.

[10] Yu Jin, Nick Duffield, Alexandre Gerber, Patrick Haffner, Subhabrata Sen, and Zhi-Li Zhang. NEVERMIND, the problem is already fixed: Proactively detecting and troubleshooting customer DSL problems. In *CONEXT*, 2010.

[11] Lin Quan, John Heidemann, and Yuri Pradkin. Trinocular: Understanding Internet reliability through adaptive probing. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2013.

[12] Philipp Richter, Ramakrishna Padmanabhan, David Plonka, Arthur Berger, and David Clark. Advancing the Art of Internet Edge Outage Detection. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2018.

[13] Ethan Katz-Basset, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, David Wetherall, and Thomas Anderson. Studying black holes in the internet with Hubble. In *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[14] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapè. Analysis of country-wide Internet outages caused by censorship. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2011.

[15] RIPE NCC. Atlas. http://atlas.ripe.net.

[16] SamKnows. http://www.samknows.com.

[17] Srikanth Sundaresan, Sam Burnett, Nick Feamster, and Walter de Donato. BISmark: A testbed for deploying measurements and applications in broadband access networks. In *Proceedings of the USENIX Annual Technical Conference*, June 2014.

[18] Yuval Shavitt and Eran Shir. DIMES: Let the Internet Measure Itself. *SIGCOMM Comput. Commun. Rev.*, 35, October 2005.

[19] Ramakrishna Padmanabhan, Patrick Owen, Aaron Schulman, and Neil Spring. Timeouts: Beware surprisingly high delay. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2015.

[20] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. A multi-perspective analysis of carrier-grade NAT deployment. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2016.

[21] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, kc claffy, and Neil Spring. Reasons dynamic addresses change. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2016.

[22] David D. Clark. The design philosophy of the DARPA internet protocols. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 1988.

[23] Gianluca Iannaccone, Chen-nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. Analysis of Link Failures in an IP Backbone. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2002.

[24] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental Study of Internet Stability and Wide-Area Backbone Failures. In *FTCS*, 1999.

[25] Ratul Mahajan, David Wetherall, and Thomas Anderson. Understanding BGP misconfiguration. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2010.

[26] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of country-wide Internet outages caused by censorship. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2011.

[27] Vern Paxson. End-to-end routing behavior in the Internet. In *IEEE/ACM Transactions on Networking*, 1997.

[28] Amogh Dhamdhere, Renata Teixeira, Constantine Dovrolis, and Christophe Diot. NetDiagnoser: Troubleshooting Network Unreachabilities Using End-to-end Probes and Routing Data. In *Proceedings of the 2007 ACM CoNEXT Conference*, 2007.

[29] Ethan Katz-Basset, Colin Scott, David R. Choffnes, Ítalo Cunha, Vytautas Valancius, Nick Feamster, Harsha V. Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. LIFEGUARD: Practical repair of persistent route failures. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2012.

[30] Umar Javed, Italo Cunha, David Choffnes, Ethan Katz-Bassett, Thomas Anderson, and Arvind Krishnamurthy. PoiRoot: Investigating the Root Cause of Interdomain Path Changes. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2013.

[31] Ritwik Banerjee, Abbas Razaghpanah, Luis Chiang, Akassh Mishra, Vyas Sekar, Yejin Choi, and Phillipa Gill. Internet Outages, the Eyewitness Accounts: Analysis of the Outages Mailing List. In *Proceedings of Passive & Active Measurement (PAM)*, 2015.

[32] Daniel Turner, Kirill Levchenko, Alex C. Snoeren, and Stefan Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2010.

[33] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2000.

[34] John P. John, Ethan Katz-Bassett, Arvind Krishnamurthy, Thomas Anderson, and Arun Venkataramani. Consensus Routing: The Internet As a Distributed System. In *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[35] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A Measurement Study on the Impact of Routing Events on End-to-end Internet Path Performance. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2006.

[36] Nate Kushman, Srikanth Kandula, and Dina Katabi. Can You Hear Me Now?!: It Must Be BGP. *ACM SIGCOMM Computer Communication Review*, 37(2):75–84, March 2007.

[37] Internet Outage Detection and Analysis (IODA). https://www.caida.org/projects/ioda/.

[38] Routeviews Project – University of Oregon. http://www.routeviews.org/.

[39] RIPE Routing Information Service. http://www.ripe.net/ris/.

[40] Sarthak Grover, Mi Seon Park, Srikanth Sundaresan, Sam Burnett, Hyojoon Kim, Bharath Ravi, and Nick Feamster. Peeking behind the NAT: an empirical study of home networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2013.

[41] Anant Shah, Romain Fontugne, Emile Aben, Cristel Pelsser, and Randy Bush. Disco: Fast, good, and cheap outage detection. In *TMA*, 2017.

[42] 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.

[43] 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 *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2013.

[44] Oded Argon, Anat Bremler-Barr, Osnat Mokryn, Dvir Schirman, Yuval Shavitt, and Udi Weinsberg. On the dynamics of IP address allocation and availability of end-hosts. *arXiv preprint arXiv:1011.2324*, 2010.

[45] Zachary S. Bischof, Fabian E. Bustamante, and Rade Stanojevic. Need, Want, Can Afford: Broadband Markets and the Behavior of Users. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2014.

[46] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security*, pages 605–620, 2013.

[47] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2008.

[48] David Adrian, Zakir Durumeric, Gulshan Singh, and J. Alex Halderman. Zippier ZMap: Internet-Wide Scanning at 10 Gbps. In *WOOT*, 2014.

[49] R. Braden, Editor. Requirements for internet hosts – communication layers. Internet Engineering Task Force Request for Comments RFC-1122, October 1989.

[50] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Aravind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, 2007.

[51] Nick Feamster, David G. Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2003.

[52] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. PlanetSeer: Internet path failure monitoring and characterization in wide-area services. In *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[53] Philip Homburg. [atlas] timeout on ping measurements. http://www.ripe.net/ripe/mail/archives/ripe-atlas/2013-July/000891.html, July 2013. Posting to the ripe-atlas mailing list.

[54] ISI ANT Lab. Internet address survey binary format description. http://www.isi.edu/ant/traces/topology/address_surveys/binformat_description.html.

[55] Matthew Luckie. Scamper: A Scalable and Extensible Packet Prober for Active Measurement of the Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, pages 239–245, 2010.

[56] Neil Spring, David Wetherall, and Tom Anderson. Scriptroute: A public Internet measurement facility. In *USENIX Symposium on Internet Technologies and Systems (USITS)*, 2003.

[57] Jeffrey Mogul. Broadcasting Internet datagrams. Internet Engineering Task Force Request for Comments RFC-919, October 1984.

[58] Landernotes. https://wiki.isi.edu/predict/index.php/LANDER:internet_address_survey_reprobing_it54c-20130524.

[59] Fred Baker. Requirements for IP version 4 routers. Internet Engineering Task Force Request for Comments RFC-1812, June 1995.

[60] Mathew J. Luckie, Anthony J. McGregor, and Hans-Werner Braun. Towards improving packet probing techniques. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2001.

[61] Ina Minei and Reuven Cohen. High-speed internet access through unidirectional geostationary satellite channels. In *IEEE Journal on Selected Areas in Communications*, 1999.

[62] Chadi Barakat, Nesrine Chaher, Walid Dabbous, and Eitan Altman. Improving TCP/IP over geostationary satellite links. In *Global Telecommunications Conference, 1999. GLOBECOM'99*, volume 1, pages 781–785, 1999.

[63] Rajiv Chakravorty, Andrew Clark, and Ian Pratt. GPRSWeb: Optimizing the web for GPRS links. In *Proceedings of the International Conference on Mobile Systems, Applications and Services (MOBISYS)*, May 2003.

[64] Stefan Saroiu, P. Krishna Gummadi, and Steven D Gribble. Measurement study of peer-to-peer file sharing systems. In *MMCN*, 2002.

[65] Jacky C. Chu, Kevin S. Labonte, and Brian N. Levine. Availability and locality measurements of peer-to-peer file systems. In *ITCom: Scalability and Traffic Control in IP Networks*, 2002.

[66] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Transactions on Networking (ToN)*, 12(2):219–232, 2004.

[67] Vyas Sekar, Yinglian Xie, Michael K Reiter, and Hui Zhang. A multi-resolution approach for worm detection and containment. In *DSN*, 2006.

[68] Andrew J. Kaizer and Minaxi Gupta. Open resolvers: Understanding the origins of anomalous open DNS resolvers. In *Proceedings of Passive & Active Measurement (PAM)*, 2015.

[69] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going wild: Large-scale classification of open dns resolvers. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2015.

[70] Yinglian Xie, Vyas Sekar, David Maltz, Michael K Reiter, Hui Zhang, et al. Worm origin identification using random moonwalks. In *Proc. of the IEEE Symposium on Security and Privacy*, 2005.

[71] Jaeyeon Jung and Emil Sit. An empirical study of spam traffic and the use of DNS black lists. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2004.

[72] MARJZ Fabian and Monrose Andreas Terzis. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, USA*, 2007.

[73] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the 16th ACM conference on Computer and communications security*, 2009.

[74] Dennis Andriesse, Christian Rossow, and Herbert Bos. Reliable recon in adversarial peer-to-peer botnets. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2015.

[75] Fail2ban. http://www.fail2ban.org/.

[76] The spamhaus project. http://www.spamhaus.org/.

[77] The cbl. http://www.abuseat.org/.

[78] Sorbs (spam and open-relay blocking system). www.sorbs.net/.

[79] Ralph Droms. RFC 2131: Dynamic host configuration protocol. Internet Engineering Task Force Request for Comments RFC-2131, March 1997.

[80] William Simpson. The Point-to-Point Protocol. Internet Engineering Task Force Request for Comments RFC-1661, July 1994.

[81] Glenn McGregor. The PPP Internet Protocol Control Protocol (IPCP). Internet Engineering Task Force Request for Comments RFC-1332, May 1992.

[82] Vladimir Brik, Jesse Stroik, and Suman Banerjee. Debugging DHCP performance. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2004.

[83] Manas Khadilkar, Nick Feamster, Matt Sanders, and Russ Clark. Usage-based DHCP lease time optimization. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2007.

[84] Ioannis Papapanagiotou, Erich M. Nahum, and Vasileios Pappas. Configuring DHCP leases in the smartphone era. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2012.

[85] Yinglian Xie, Fang Yu, Kannan Achan, Eliot Gillum, Moises Goldszmidt, and Ted Wobber. How dynamic are IP addresses? In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2007.

[86] Giovane CM Moura, Carlos Ganán, Qasim Lone, Payam Poursaied, Hadi Asghari, and Michel van Eeten. How dynamic is the isps address space? towards internet-wide dhcp churn estimation. *IFIP*, 2015.

[87] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2009.

[88] Martin Casado and Michael J. Freedman. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[89] Ramakrishna Padmanabhan, Zhihao Li, Dave Levin, and Neil Spring. UAv6: Alias resolution in IPv6 using unused addresses. In *Proceedings of Passive & Active Measurement (PAM)*, 2015.

[90] Thomas Narten, Richard Draves, and Suresh Krishnan. Privacy extensions for stateless address autoconfiguration in ipv6. Internet Engineering Task Force Request for Comments RFC-4941, September 2007.

[91] David Plonka and Arthur Berger. Temporal and spatial classification of active IPv6 addresses. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2015.

[92] About RIPE Atlas: FAQ: How does the probe connect to the Internet? https://atlas.ripe.net/about/faq/.

[93] Philip Homburg. NTP measurements with RIPE Atlas. https://labs.ripe.net/Members/philip_homburg/ntp-measurements-with-ripe-atlas, February 2015.

[94] RIPE NCC. RIPE atlas probe archive. https://atlas.ripe.net/api/v1/probe-archive/.

[95] RIPE NCC. RIPE atlas connection logs url format. https://atlas.ripe.net/probes/⟨prb_id⟩/connection-history/⟨yyyy⟩/⟨mm⟩/.

[96] Routeviews prefix to as mappings dataset (pfx2as) for ipv4 and ipv6. https://www.caida.org/data/routing/routeviews-prefix2as.xml.

[97] RIPE NCC. Built-in measurements. https://atlas.ripe.net/docs/built-in/.

[98] Zwangstrennung (Forced IP address change). https://de.wikipedia.org/wiki/Zwangstrennung.

[99] RIPE NCC. Become a ripe atlas probe host. https://atlas.ripe.net/get-involved/become-a-host/.

[100] RIPE NCC Staff. RIPE Atlas: A global internet measurement network. *Internet Protocol Journal*, 18(3), September 2015.

[101] RIPE NCC. Technical updates. https://atlas.ripe.net/resources/announcements/.

[102] Gerald van Belle, Patrick J. Heagerty, Lloyd D. Fischer, and Thomas S. Lumley. *Biostatistics: A Methodology for the Health Sciences (Second Edition)*. John Wiley & Sons, 2004.

[103] Hang Guo and John S. Heidemann. Detecting ICMP rate limiting in the internet. In *Proceedings of Passive & Active Measurement (PAM)*, 2018.

[104] Comcast outage on Sep 13 2017 in the Outages Mailing List. https://puck.nether.net/pipermail/outages/2017-September/010754.html.

[105] National Hurricane Center Tropical Cyclone Report: Hurricane Irma. https://www.nhc.noaa.gov/data/tcr/AL112017_Irma.pdf.

[106] Northeast Storm Undergoes Bombogenesis, Bringing 70 MPH Gusts, Almost 350 Reports of Wind Damage, Flooding — The Weather Channel. https://weather.com/forecast/regional/news/2017-10-30-northeast-storm-damaging-winds-flooding.

[107] October 29-30, 2017 damaging winds, heavy rainfall & flooding. https://www.weather.gov/aly/October29-302017.

[108] More than 1 million power outages in the Northeast after blockbuster fall storm - The Washington Post. https://www.washingtonpost.com/news/capital-weather-gang/wp/2017/10/30/over-one-million-power-outages-in-the-northeast-after-blockbuster-fall-storm/.

[109] Line Of Storms Moves Through Oklahoma. http://www.newson6.com/story/36651816/tornado-watch-in-effect-for-ne-oklahoma.

[110] U.S. Government. CFR part 4 section 4.9: Outage reporting requirements threshold criteria.

[111] Edmond W. W. Chan, Xiapu Luo, Waiting W. T. Fok, Weichao Li, , and Rocky K. C. Chang. Non-cooperative diagnosis of submarine cable faults. In *Proceedings of Passive & Active Measurement (PAM)*, 2011.

[112] Tomasz Bilski. Disaster's impact on Internet performance – case study. In *CCIS*, 2009.

[113] John Heidemann, Lin Quan, and Yuri Pradkin. A preliminary analysis of network outages during hurricane Sandy. Technical report, USC/ISI, 2012.

[114] Gianluca Iannaccone, Chen nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. Analysis of link failures in an IP backbone. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2002.

[115] Frank B. Jewett. The modern telephone cable. In *Proceedings of 26th annual convention of the American Institute of Electrical Engineers*, 1909.

[116] W. T. Smith and W. L. Roberts. Design and characteristics of coaxial cables for Community Antenna Television. *IEEE Transactions on Communication Technology*, 1966.

[117] D.C. Hogg and Ta-Shing Chu. The role of rain in satellite communications. *PROC-IEEE*, 1975.

[118] Helmut Bölcskei, Arogyaswami J. Paulraj, K. V. S. Hari, Rohit U. Nabar, and Willie W. Lu. Fixed broadband wireless access: State of the art, challenges, and future directions. *IEEE Communications Magazine*, 2001.

[119] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2008.

[120] RIPE NCC. RIPE Atlas. http://atlas.ripe.net.

[121] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Broadband Internet performance: A view from the gateway. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 2011.

[122] Aaron Schulman and Neil Spring. Pingin' in the rain. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2011.

[123] NOAA. Automated surface observing system (ASOS). https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/automated-surface-observing-system-asos. ftp://ftp.ncdc.noaa.gov/pub/data/noaa/.

[124] NOAA. State of the climate report: Hurricanes and tropical storms. https://www.ncdc.noaa.gov/sotc/.

[125] Cisco. GS7000 DOCSIS status monitor transponder installation and operation guide, 2011. https://www.cisco.com/c/dam/en/us/td/docs/video/access_edge/Nodes/GS7000/4037424_A.pdf.

[126] Alpha Technologies. Installation and technical manual DOCSIS HMS embedded transponder, 2004.

[127] Xiaoliang Zhao, Daniel Massey, Mohit Lad, and Lixia Zhang. On/off model: A new tool to understand bgp update burst. Technical report, University of California, Los Angeles, 2004.

[128] 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.

[129] G. Comarela, G. Gürsun, and M. Crovella. Studying interdomain routing over long timescales. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2013.

[130] J. M. Wallace. Diurnal variations in precipitation and thunderstorm frequency over the conterminous United States. *Monthly Weather Review*, 1975.

[131] Carolyn Orr. A look at how and why North Dakota became a leader in deployment of fiber optic Internet. http://www.csgmidwest.org/policyresearch/0616-fiber-optic-North-Dakota.aspx, Jun 2016.

[132] Chris Hill. 23 state DOTs spent more than $1 billion on snow, ice maintenance this winter. https://www.equipmentworld.com/23-state-dots-spent-more-than-1-billion-on-snow-ice-maintenance-this-winter/, May 2015.

[133] Remote Access (DynDNS). http://dyn.com/remote-access/.

[134] the Dynamic DNS page - FreeDNS. https://freedns.afraid.org/dynamic/.

[135] No-IP: Free Dynamic DNS. www.noip.com.

[136] Open resolver project. http://openresolverproject.org/.

[137] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. On measuring the client-side DNS infrastructure. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2013.

[138] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. Exit from hell? reducing the impact of amplification ddos attacks. In *USENIX Security Symposium*, 2014.

[139] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. Assessing DNS vulnerability to record injection. In *Proceedings of Passive & Active Measurement (PAM)*, 2014.

[140] Youndo Lee and Neil Spring. Identifying and analyzing broadband internet reverse DNS names. In *CONEXT*, 2017.

[141] Ramakrishna Padmanabhan. We can find shared IP addresses. https://blog.apnic.net/2018/03/05/can-find-shared-ip-addresses/.

[142] Ramakrishna Padmanabhan. Analyzing static, dynamic, and gateway IPv4 addresses. In *AIMS 2017: Workshop on Active Internet Measurements*, 2017.