Timeouts: Beware Surprisingly High Delay

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

Patrick Owen
University of Maryland
powen@cs.umd.edu

Aaron Schulman
Stanford University
aschulm@stanford.edu

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

ABSTRACT

Active probing techniques, such as ping, have been used to detect outages. When a previously responsive end host fails to respond to a probe, studies sometimes attempt to confirm the outage by retrying the ping or attempt to identify the location of the outage by using other tools such as traceroute. The latent problem, however, is, how long should one wait for a response to the ping? Too short a timeout risks confusing congestion or other delay with an outage. Too long a timeout may slow the process and prevent observing and diagnosing short-duration events, depending on the experiment's design.

We believe that conventional timeouts for active probes are underestimates, and analyze data collected by Heidemann et al. in 2006–2015. We find that 5% of pings from 5% of addresses take more than 5 seconds. Put another way, for 5% of the responsive IP addresses probed by Heidemann, a false 5% loss rate would be inferred if using a timeout of 5 seconds. To arrive at this observation, we filtered artifacts of the data that could occur with too-long a timeout, including responses to probes sent to broadcast addresses. We also analyze ICMP data collected by Zmap in 2015 to find that around 5% of all responsive addresses observe a greater than one second round-trip time consistently. Further, the prevalence of high round trip time has been increasing and it is often associated with the first ping, perhaps due to negotiating a wireless connection. In addition, we find that the Autonomous Systems with the most high-latency addresses are typically cellular. This paper describes our analysis process and results that should encourage researchers to set longer timeouts when needed and report on timeout settings in the description of future measurements.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and WideArea Networks—Internet

IMC'15, October 28–30, 2015, Tokyo, Japan.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3848-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2815675.2815704.

Keywords

Timeouts; ICMP Echo; Ping; Outages; Reachability; Outage Detection; Maximum Segment Lifetime

1. INTRODUCTION

Active probes, such as the echo requests sent by ping, can be used to study network reliability [10,14,18,21,23]. A path to a destination is working if the ping is successful: an echo request solicits an echo response. An outage is detected if a previously responsive destination stops responding to successive probes, using multiple probes because infrequent loss is expected in the Internet [17]. Each study then applies a different approach to confirm or diagnose the outage.

Unfortunately, the time one should wait for a response is not well understood. Protocols like TCP and DNS use timeouts near three seconds, and various tools use comparable thresholds: iPlane [14] uses 2 seconds with one retry, Trinocular [18] uses 3 seconds, and Scriptroute [22] defaults to 3 second timeouts. However, TCP and DNS are both able to tolerate longer delays because the timeout is merely a trigger for retransmission: both give up on the connection much later. In contrast, network measurements that timeout quickly have the advantage of being responsive—they may send follow up probes to explore a potential outage—but a disadvantage in that these detected losses, and ultimately outages, may not be real.

In this paper, we try to find a good timeout for active probing. We begin by studying ping latencies from Internet-wide surveys [7] conducted by ISI, including 9.64 billion ICMP Echo Responses from 4 million different IP addresses in 2015. The probing scheme for this survey sets a timeout threshold of 3 seconds [7], although this timeout appears to vary in practice, and only matches responses that arrive before this timer fires: we call these responses *survey-detected responses.* Survey-detected responses include a microsecond-precise round-trip time. When an echo request does not receive a response before the timeout, it is recorded in the data with a timestamp in seconds. When an echo response is received that does not match an echo request that has not yet timed out, that response is also recorded in the data with a timestamp in seconds. Thus it is possible to re-process the data to identify echo responses that took longer than the timeout to arrive. We term such responses *unmatched responses*, and can determine a round trip time precise only to seconds.

We classify unmatched responses into three categories: (a) *delayed responses* potentially caused by congestion, (b) responses that were triggered by later requests sent to broadcast addresses (*broadcast responses*), and (c) *duplicate responses*, some of which appear consistent with denial of service attacks. Since broadcast responses and duplicate responses do not contribute to the latency analysis, we term them *unexpected responses* and remove them with filters. We then *verify* 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 explain 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. The classification process reveals that the Autonomous Systems with the most high latency addresses are cellular.

This paper is organized as follows. We discuss related work, primarily as a means of motivating our study by describing prior timeouts, in Section 2. We describe the ISI survey dataset and our methods of extracting high latency despite a short timeout in Section 3. Section 4 provides the key results: how long a timeout must be to capture a high percentage of responses from a high percentage of hosts. Section 5 addresses doubts about whether these latencies are real, and Section 6 focuses on identifying the networks and behaviors responsible for high latencies. We conclude in Section 7 with our recommendations.

2. IMPORTANCE OF PROBE TIMEOUTS

In this section, we describe why it is important to choose an appropriate timeout for active probes, especially when used for outage detection. We also describe measurement studies with particular attention to what timeouts were used and how those timeouts were chosen.

2.1 Selecting a timeout

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.

When a probe experiences a timeout, it is generally assumed that either the probe is lost or the end-host is no longer reachable. For most active probing systems, any timed out active probes are followed up with retransmissions to increase the confidence that a lack of response is due to a lack of reachability and not loss. These followup probes will also have a timeout that is generally the same as the first attempt.

Studies on Internet outages and connectivity problems rely on these probe timeouts to indicate that hosts are no longer reachable. However, non-responses to active probes within a timeout can occur for other reasons than the host being offline. 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.

Even for studies that don't focus upon outages, selecting a good timeout is important. For instance, in the ISI surveys we study, most probes solicit no responses. To the best of our knowledge, this is the first paper that investigates this broad lack of responses to see if researchers are simply using timeouts that are too short.

2.2 Timeouts used in outage and connectivity studies

Outage detection systems such as Trinocular [18] and Thunderping [21] tend to use a 3 second timeout for active probes because it is the default TCP SYN/ACK timeout [3]. Trinocular probes all /24s on the Internet [18]. It does so by sending ICMP echo requests to a few addresses in all /24 address blocks and analyzes responses to detect outages on the block level. Trinocular performs adaptive retransmission and sends up to 15 additional probes to an address block before declaring an outage for that block. Thunderping [21] sends ICMP echo requests to IP addresses that are likely to be subject to severe weather from multiple vantage points periodically and detects outages when all vantage points fail to receive a response. It executes its probing with Scriptroute [22], where each probe has a 3 second timeout. Thunderping retransmits probes ten times before declaring a host is unresponsive.

Internet performance monitoring systems use a wide range of probe timeouts. On the shorter side, iPlane [14] and Hubble [10] 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. [6] 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 [23] 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 [19] probing hardware sends continuous pings to various hosts on the Internet to observe connectivity. The timeout for their ICMP echo requests is 1 second [8]. The SamKnows probing hardware uses a 3 second timeout for ICMP echo requests sent during loaded intervals [20].

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. 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.

3.1 Raw ISI survey data

ISI has conducted Internet wide surveys [7] since 2006. Precise details can be found in Heidemann et al. [7], and technical details of the data format online [9], but we present a brief overview here.

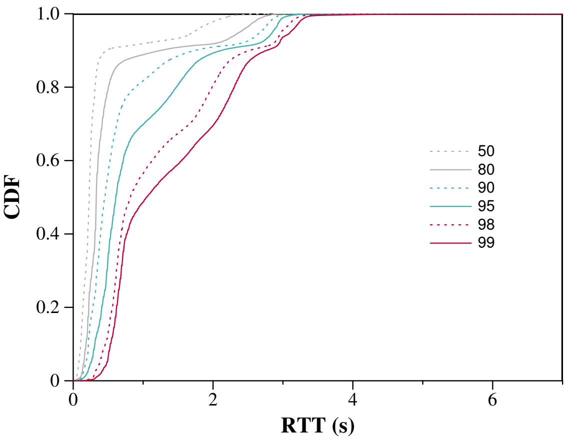


Figure 1: CDF of percentile latency of survey-detected responses per IP address: Each point represents an IP address and each curve represents the percentile from that IP address's response latencies. The slope of the latency percentiles increases around the 3 second mark, suggesting that ISI's prober timed out responses that arrived after 3 seconds.

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 [5] and additional experiments including more frequent probing with Scamper [13] and Scriptroute [22].

3.2 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 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.

3.3 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 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 5 below.

3.3.1 *Broadcast responses*

The dataset contains several instances where a ping to a destination times out, but is closely followed by an un-

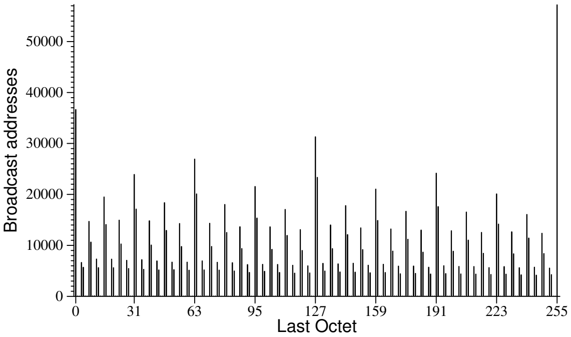


Figure 2: Broadcast addresses that solicit responses in Zmap: Broadcast addresses usually have last octets whose last N bits are either 1 or 0 (where N > 1).

matched 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 [16]. The broadcast address in a network should be an address that is unlikely to be assigned to a real host [16], 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 [3], 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 [5] 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 addresses 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 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 within the same /24 prefix for each unmatched response. We then extract the last octet of the most recently probed address. Figure 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 2.

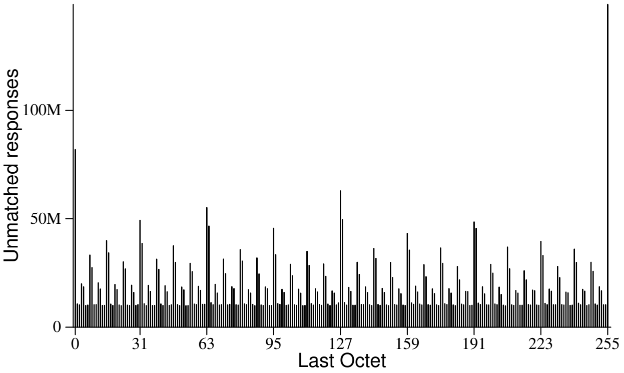


Figure 3: Number of unmatched responses that followed a probe sent to address with last octet X. Last octets with last N bits ending in 0s and 1s (where N > 1) observe spikes, likely caused by broadcast responses. Not all unmatched responses are caused by broadcast responses, however, since there exist roughly 10M unmatched responses distributed evenly across all last octets.

If left in the data, broadcast responses could yield substantial latency overestimates in the following, common, scenario, which we illustrate in Figure 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 4. For every unmatched response with a latency of at least 10 seconds, the filter checks if 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

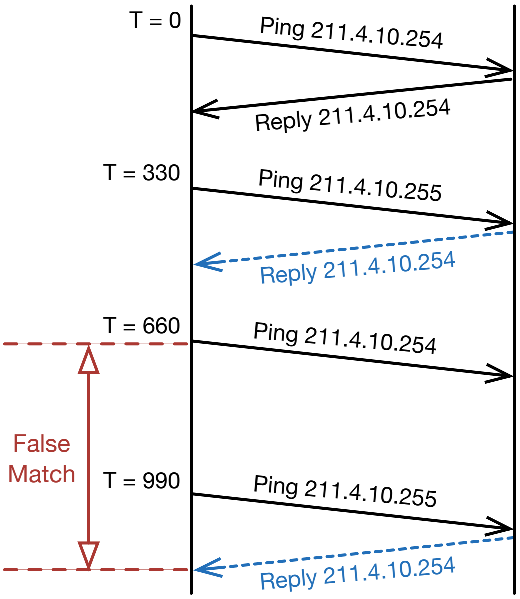


Figure 4: We filter broadcast responses since they can lead to the inference of false latencies. This figure illustrates a potential incorrect match caused by a broadcast response. Echo requests sent to the broadcast address 211.4.10.255 at T = 330 and T = 990 seconds solicit responses from 211.4.10.254. 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.

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%.

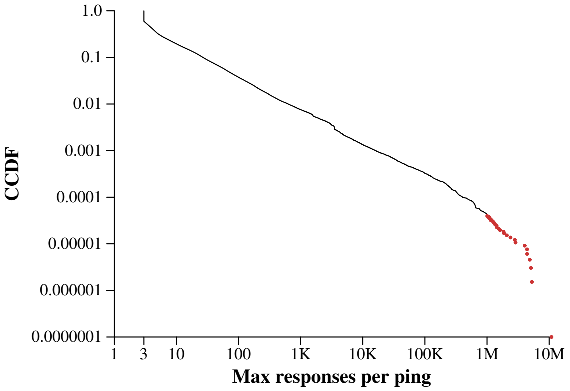


Figure 5: Maximum number of responses received for a single echo request, for IP addresses that sent more than 2 responses to an echo request. The red dots indicate instances where addresses responded to a single echo request with more than 1M echo responses. We believe that these are caused by DoS attacks.

3.3.2 *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, 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 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 [5]. 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 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.

4. 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

	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 1: Adding unmatched responses to survey-detected responses

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 the ISI dataset; we verify that these are real in Section 5 and examine causes in Section 6.

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

4.2 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 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.

		% 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 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).

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 6.

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.

5. 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.

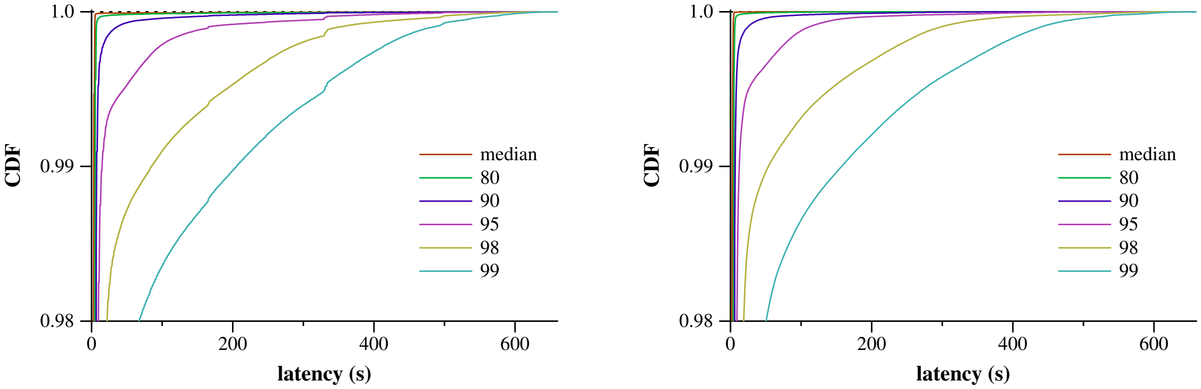
5.1 Are high latencies observed by other probing schemes?

Some of the latencies in Table 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 [5]. 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. Typically, scans were per-



(a) Before filtering

(b) After filtering

Figure 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.

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: Zmap scan details: For each Zmap scan in Figure 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.

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

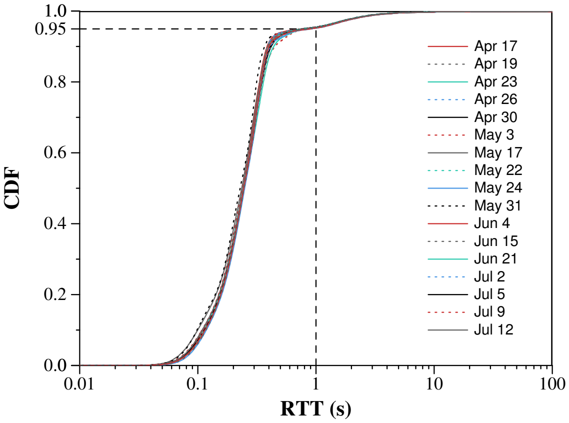


Figure 7: Distribution of RTTs for all Zmap scans performed in 2015. Around 5% of addresses have latencies greater than 1s in each scan, and 0.1% of addresses observed latencies in excess of 75s.

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 [13], 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 [13] 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

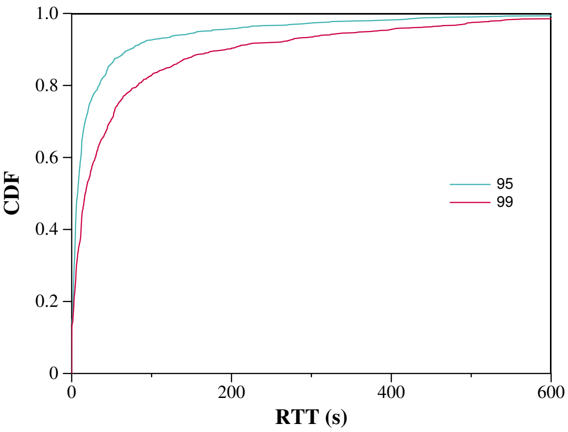


Figure 8: Confirmation of high latency: Percentile latency per IP address for 2000 randomly chosen IP addresses from ISI's 2011 - 2013 surveys that had $> 5\%$ of pings with latencies 100s and above. Each point represents an IP address and the lines represent the percentile latency from that IP address. 17% of them continue to observe 1% of their pings with latencies > 100s.

later analyses, we ran tcpdump simultaneously and matched responses to sent packets separately.

Of the 2000 addresses, 1244 responded to our probes. Figure 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 6.

Nevertheless, Figure 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.

5.2 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".

In this section, we look at summary metrics of each of the surveys. In Figure 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) and it54w (20130430) were flagged by ISI as having high latency variation due to a software error [11].

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.

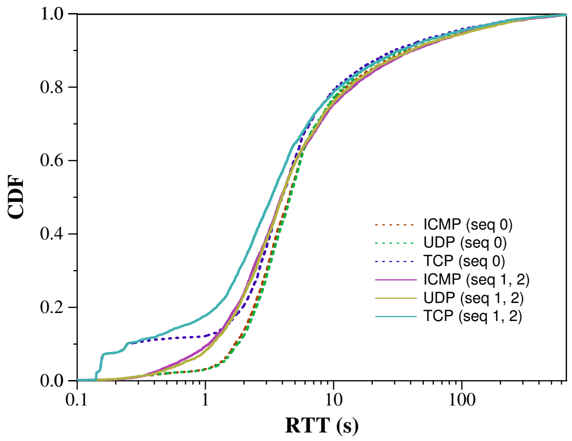


Figure 10: 98th percentile RTTs associated with high-latency IP addresses using different probe protocols. The first probe of a triplet (seq 0) often has a higher latency than the rest; TCP probes appear to have a similar distribution except for firewall-sourced responses.

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.

5.3 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 [1,12], 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.

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

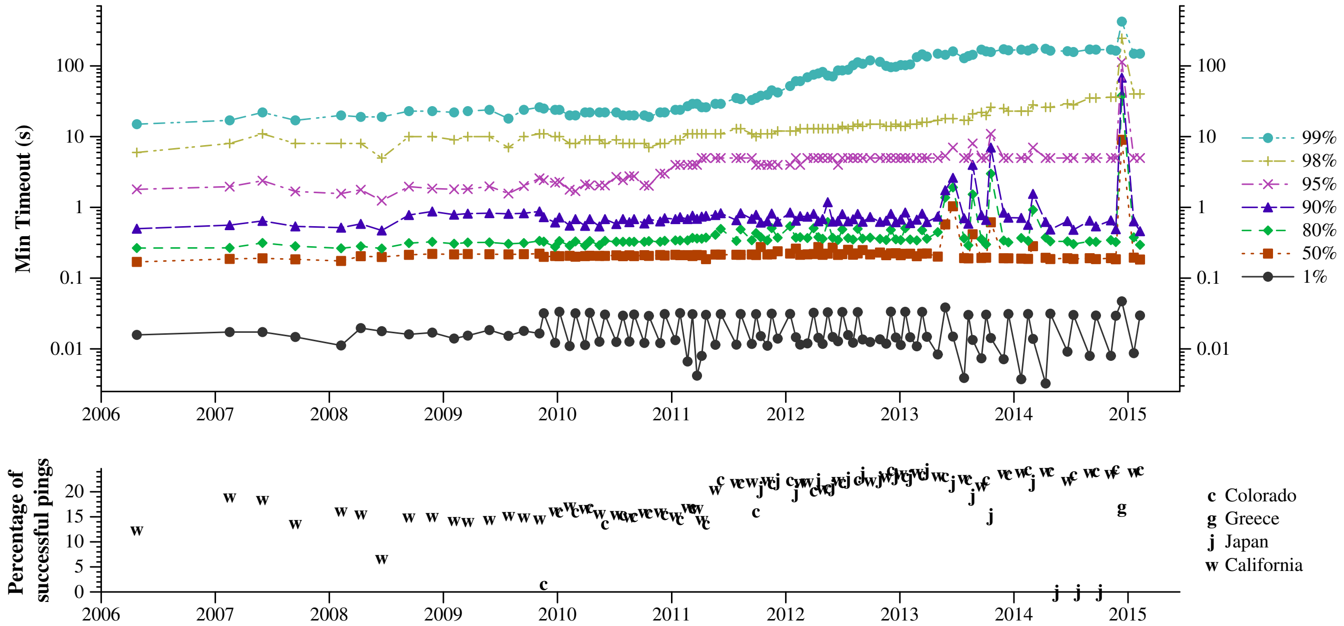


Figure 9: Top: Minimum timeout required to capture the c^{th} percentile latency sample from the c^{th} percentile address in each survey, organized by time. Each point represents the timeout required to capture, e.g., 95% of the responses from 95% of the addresses. The 1% line is indicative of the minimum. 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.

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 [13] 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 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 6.3.

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.

5.4 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.

6. 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.

6.1 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

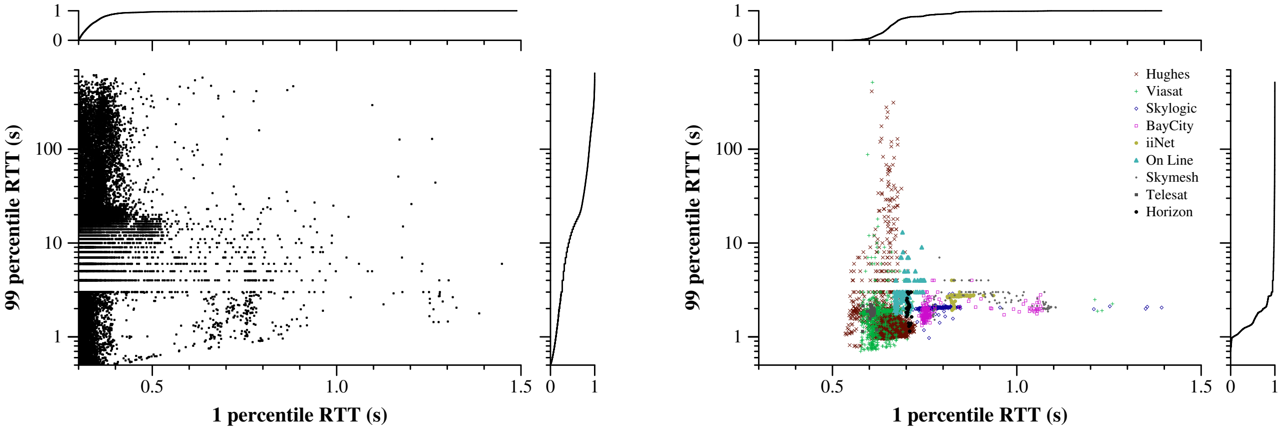


Figure 11: Scatterplot of 1st and 99th percentile latencies for addresses with high values of both in survey IT63c; Left omits satellite-only ISPs; Right includes only satellite-only ISPs.

a satellite and back, leading to about 125 ms of one way delay [2, 15]. 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 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 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.

6.2 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, 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. 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 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 *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 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

		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 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.

	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 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.

		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 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.

in the June and July scans. Thus, the Autonomous Sytems with the most turtles also remain consistent over time.

Table 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 4, Table 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 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.

6.3 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 [4], but showing quantitatively more significant delay.

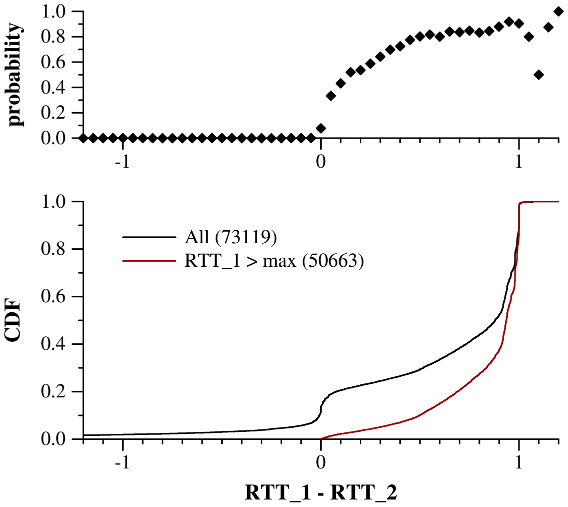


Figure 12: Bottom: Difference between initial latency and second probe latency; values around 1 indicate that both responses arrive at about the same time, values near zero indicate that the RTTs were about the same. The second line includes only those where $RTT_1 > max(RTT_2 \ldots RTT_n)$. Top: The probability that, given $RTT_1 - RTT_2$ on the x-axis, that $RTT_1 > max(RTT_2 \ldots RTT_n)$.

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, $RTT_2 \ldots RTT_n$, 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, $median(RTT_2 \ldots RTT_n) < RTT_1 < max(RTT_2 \ldots RTT_n)$, 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 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 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 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.

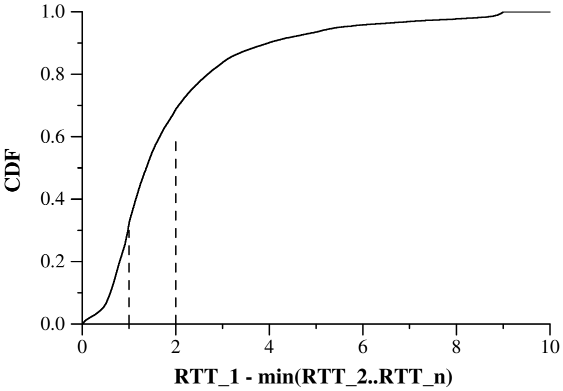


Figure 13: Difference between initial latency and observed minimum. The typical setup time is below four seconds.

We infer that a measurement approach that sent a second probe after one second could detect this behavior. The top graph of Figure 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, 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 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 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 re-

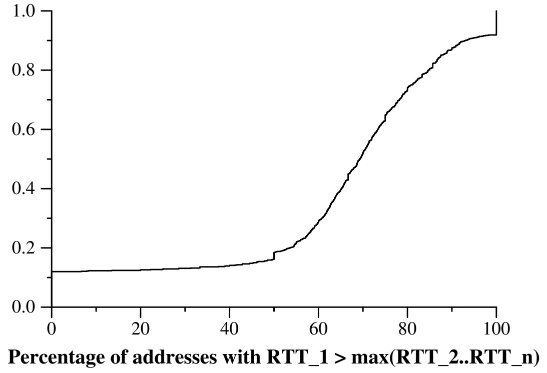


Figure 14: Percentage of addresses in a /24 prefix showing a drop from the initial to the maximum.

sponsive addresses within each prefix that dropped from the initial ping to the maximum of the rest in Figure 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.

6.4 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.

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

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 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.*

6.5 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.

7. CONCLUSIONS 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.

Our initial hypothesis was that it would be a simple matter to confirm that widely used timeout values would be adequate for studying outages, or failing that, that one or two additional seconds would be enough. However, 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 2. We plan to use 60 seconds when we need a timeout, and avoid timeouts otherwise.

Acknowledgments

We would like to express our sincere appreciation to the authors of the ISI Internet survey for both publishing their data and designing their data collection method to collect the unmatched responses that enabled this analysis.

We also would like to thank Zakir Durumeric for incorporating our changes into https://github.com/zmap/zmap/blob/master/src/probe_modules/module_icmp_echo_time.c in order to support explicit matching of responses and calculating round trip times in the stateless Zmap.

This research was supported in part by ONR grant N00173-13-1-G001.

8. REFERENCES

[1] Fred Baker. Requirements for IP version 4 routers. IETF RFC-1812, June 1995.

[2] 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.

[3] R. Braden, Editor. Requirements for internet hosts – communication layers. IETF RFC-1122, October 1989.

[4] Rajiv Chakravorty, Andrew Clark, and Ian Pratt. GPRSWeb: Optimizing the web for GPRS links. In *MOBISYS*, May 2003.

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

[6] Nick Feamster, David G. Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *ACM SIGMETRICS*, 2003.

[7] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and Survey of the Visible Internet. In *IMC*, 2008.

[8] 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.

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

[10] 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 *NSDI*, 2008.

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

[12] Mathew J. Luckie, Anthony J. McGregor, and Hans-Werner Braun. Towards improving packet probing techniques. In *IMW*, pages 145–150, San Francisco, CA, November 2001.

[13] Matthew Luckie. Scamper: A Scalable and Extensible Packet Prober for Active Measurement of the Internet. In *IMC*, pages 239–245, 2010.

[14] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Aravind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *OSDI*, Seattle, WA, November 2006.

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

[16] Jeffrey Mogul. Broadcasting Internet datagrams. IETF RFC-919, October 1984.

[17] Vern Paxson. End-to-end routing behavior in the Internet. In *ACM SIGCOMM*, pages 25–38, Palo Alto, CA, August 1996.

[18] Lin Quan, John Heidemann, and Yuri Pradkin. Trinocular: Understanding Internet Reliability Through Adaptive Probing. In *ACM SIGCOMM*, pages 255–266, 2013.

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

[20] SamKnows. Test methodology white paper, 2011.

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

[22] Neil Spring, David Wetherall, and Thomas Anderson. Scriptroute: A public Internet measurement facility. In *USITS*, pages 225–238, Seattle, WA, March 2003.

[23] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. PlanetSeer: Internet path failure monitoring and characterization in wide-area services. In *OSDI*, San Francisco, CA, December 2004.