Timeouts: Beware surprisingly high delay

Ramakrishna Padmanabhan Patrick Owen Aaron Schulman Neil Spring

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 from broadcast addresses. We find that the prevalence of high round trip time has been increasing, and that it is often associated with the first ping, perhaps due to negotiating a wireless connection. 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.

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, 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 time-out 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 3s [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 to selected hosts, 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.

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 for outage detection. We also describe related work and what timeout they used, and when the information is available, how they selected it.

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 the response arrives, if the probe even solicits a response.

For some outage studies, such as 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 in outage and connectivity studies

Outage detection systems tend to use a three second timeout for active probes because it is the default TCP SYN/ACK timeout [3]. Trinocular sends probes to all /24s on the Internet [18]. It does so by sending ICMP echo request probes to a few addresses in all /24 IP 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 request probes 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. Thunderping executes its probing with Scriptroute [22], where each probe has a three second timeout. Thunderping retransmits probes ten times before declaring a host is unresponsive.

Internet performance monitoring systems use a wide range of timeouts. On the shorter side, iPlane [14] and Hubble [10] use a two second timeout, and retransmit once for each lost probe. On the longer side, Feamster et al. [6] used a one hour timeout after each probe. However, Feamster 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 probes is one second [8]. The SamKnows probing hardware uses a three second timeout for ICMP echo probes 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.

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*, and focus only on those in this section.

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.

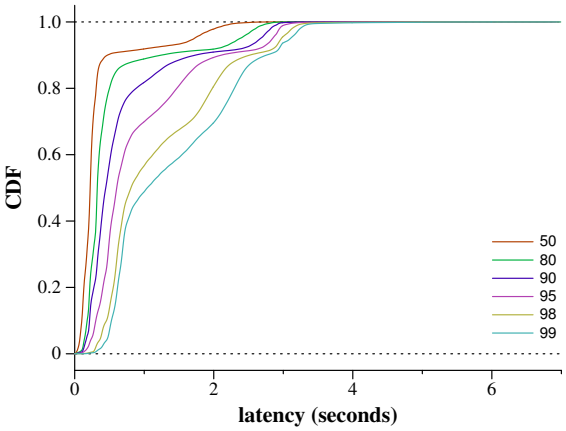


Figure 1: CDF of percentile latency of survey-detected responses per IP address: Each point represents an IP address and each color represents the percentile from that IP address's response latencies. The slope of the latency percentiles increases around the 3 second mark, suggesting that ISI's prober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

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 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

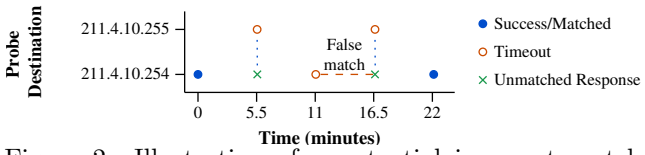


Figure 2: Illustration of a potential incorrect match due to broadcast responses. Echo requests sent to the broadcast address 211.46.104.255 at time 5.5 and 16.5 result in responses from 211.46.104.254. When a timeout occurs for a request sent directly to 211.46.104.254 at time 11 minutes, we would falsely connect that request to the response at 16.5.

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 identify and 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*

Network prefixes often include a broadcast address, where one address within a subnet represents all devices connected to that prefix [16]. 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 its own. We call these responses *broadcast responses*. No device will send an echo response with the source address that is the broadcast destination of the echo request.

If left in the data, broadcast responses could yield substantial latency overestimates in the following, common, scenario, which we illustrate in Figure 2. Assume that the echo request sent to an address 211.46.104.254 is lost and that the device is configured to respond to broadcast pings. The echo request sent to 211.46.104.254 could then be matched to the response to the request sent to 211.46.104.255, the broadcast address of the enclosing prefix. This would lead to a latency based on the interval between probing 211.46.104.254 and 211.46.104.255, 330 seconds

We examine how many of the unmatched responses in the dataset are broadcast responses. For each unmatched response, we find the last probed address within the same /24 address block and extract its last octet. Figure 3 shows the distribution of unmatched responses across these last octets. We find that last octets that can be represented as $2^n - 1$ are most likely to see an unmatched response after an echo request to them times out. We also observe smaller spikes for last octets that can be represented as 2^n. This is because broadcast addresses typically end in all 0s or all 1s. A last octet of 255 and 0 have particularly large spikes, corresponding to the possible broadcast addresses of a /24 subnet.

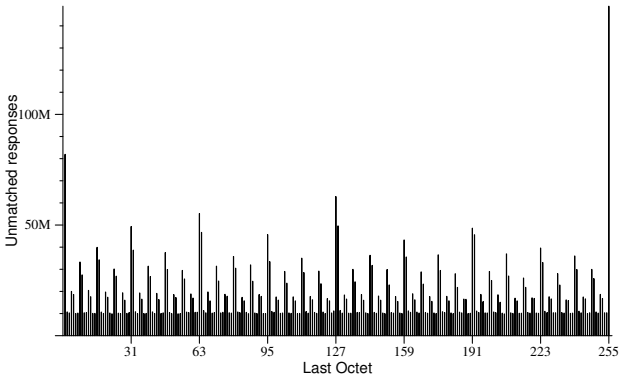


Figure 3: Number of unmatched responses that followed a probe sent to address with last octet X. Addresses with last octets that can be represented as $2^n - 1$ most often see an unmatched response after a request to them times out. Addresses with last octet that can be represented as 2^n also observe spikes, though smaller.

ISI's probing scheme sends probes to each address in a /24 address block in a nonrandom sequence. The last octet of the probed IP address generally has its nth bit (counting from 0) swapped every 2^n probes on the /24 block. We filter broadcast responses by looking for unmatched responses with similar latency that repeat periodically. 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.

In detail, the filter considers unmatched responses with a latency of at least 10 seconds and checks whether they occurred right before another response of a similar latency (within 2 seconds). We take an exponentially weighted moving average of the ratio of times this occurs with $\alpha = 0.01$. If the maximum of this moving average is greater than 0.2, the IP address is marked as one that responds to probes to broadcast addresses, and it is filtered out. This filter is meant to be selective for the specific pattern of responses, without being so selective that it fails to classify broadcast addresses in the presence of loss, and works well.

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.

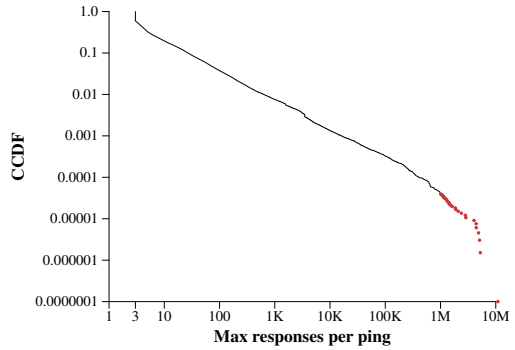


Figure 4: Maximum number of responses received for a single echo request, for IP addresses that sent more than 2 responses to an echo request. The red dots indicate instances where addresses responded to a single echo request with more than 1M echo responses. We believe that these are caused by DoS attacks.

Figure 4 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 4. 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 dataset 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 the ISI dataset; we verify that these are real in Section 5 and examine causes in Section 6

4.1 Incorporating unmatched responses

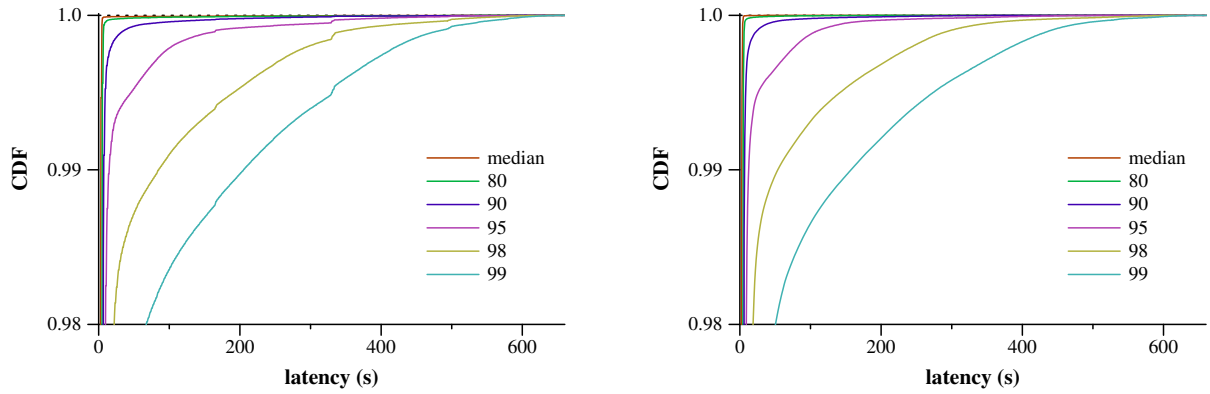
	Packets	Addresses
Survey-detected	9,644,670,150	4,008,830
Naive matching	9,768,703,324	4,008,830
Broadcast responses	33,775,148	9,942
Duplicate responses	67,183,853	20,736
Survey + Delayed	9,667,744,323	3,978,152

Table 1: Adding unmatched responses to survey-detected responses

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 5 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.



(a) Before filtering

(b) After filtering

Figure 5: 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.

	% of pings						
% of addresses	1%	50%	80%	90%	95%	98%	99%
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).

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.

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 5s. Restated, at least 5% of pings from 5% of addresses have latencies higher than 5s. Thus, even setting a timeout as high as 5s will infer a false loss rate of 5% for these addresses.

At the extreme, we see 1% of pings from 1% of addresses having latency above 145s! 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 might be spurious, that they might be a product of errors in a particular data set, that they might derive from discrimination against ICMP, or that they might be caused by ISI's data collection or sampling methodology.

5.1 Is it fleeting?

We confirm the existence of IP addresses that respond to echo requests with very high latency. A preliminary analysis of the survey dataset revealed 20,095 IP addresses that had at least 5% of their pings with latencies 100s 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 analyze the responses.

Figure 6 shows the percentile latency per IP address. We continued to observe very high latencies for these addresses.

5.2 Is it a particular survey or vantage point?

ISI survey data are collected from four vantage points

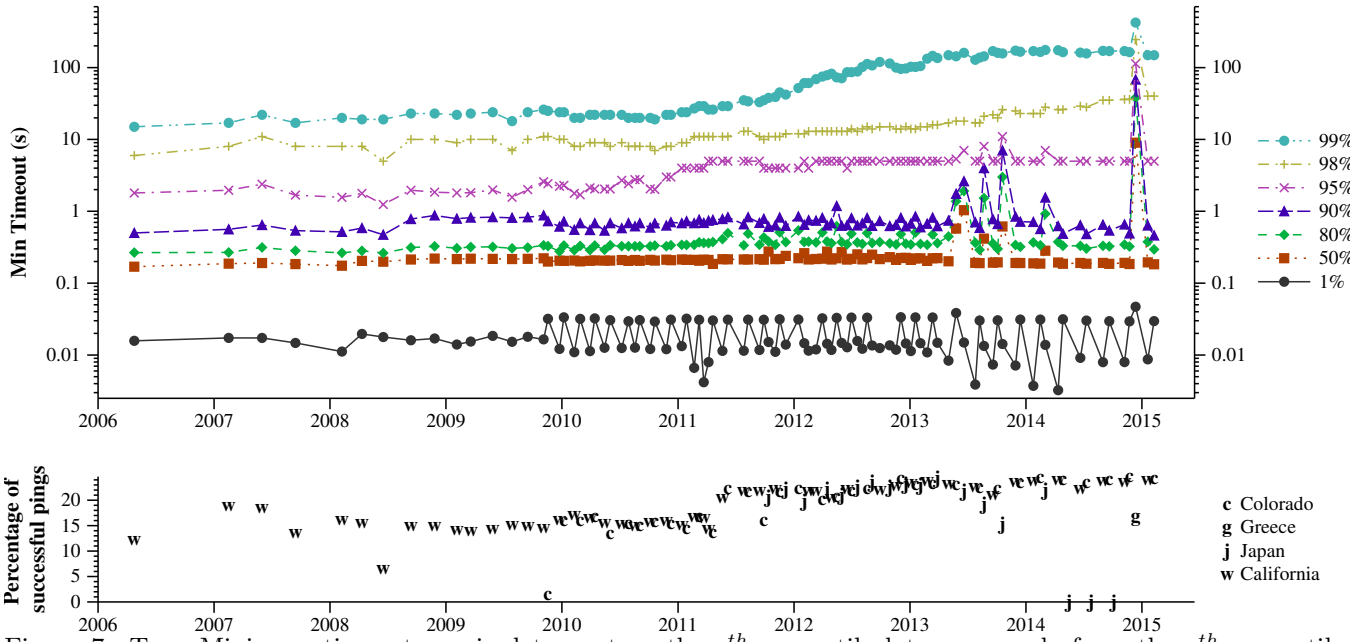


Figure 7: Top: Minimum timeout required to capture the c^{th} percentile latency sample from the c^{th} percentile address in each survey, organized by time. Each point represents the timeout required to capture, e.g., 95% of the responses from 95% of the addresses. The 1% line is indicative of the minimum. 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.

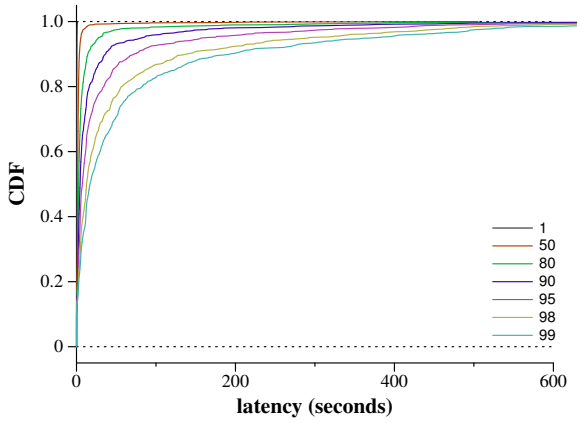


Figure 6: Confirmation of high latency: Percentile latency per IP address. Each point represents an IP address and the lines represent the percentile latency from that IP address.

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 7, 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.

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

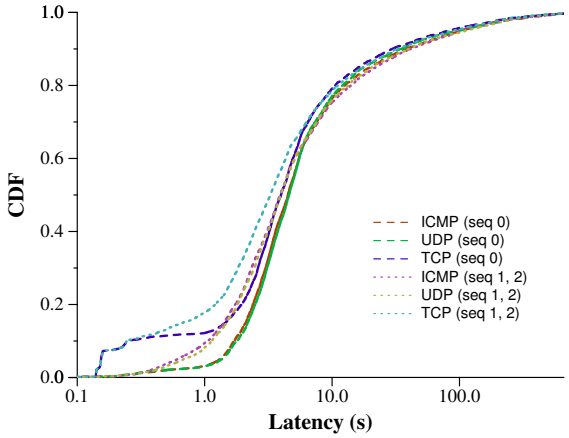


Figure 8: RTTs associated with high-latency IP addresses using different probe protocols. The first probe of a triplet (seq 0) often has a higher latency than the rest; TCP probes appear to have a similar distribution except for firewall-sourced responses.

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. However, one can imagine firewalls or similar devices that would interfere specifically with ICMP.

To consider 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 [13] to send the probe stream to each of the candidate addresses. Note that scamper uses a 2 second 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).

Figure 8 shows the distribution of measurements using each of the three protocols to these high-latency addresses. 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 6.2.

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.

6. WHY DO PINGS TAKE SO LONG?

In this section, we attempt to determine what characterizes the high-latency addresses.

6.1 Are satellites involved?

A reasonable hypothesis is that satellite links, widely known for their necessarily high latency, would be culpable for requiring high timeout values. Transmissions via geosynchronous satellite must transit 35,786km to a satellite and back, leading to about 250 ms of one way delay [2, 15]. Another 250 ms for the return trip yields a minimum of 500ms.

In Figure 9, we consider whether addresses in primarily satellite ISPs show high 99th percentile latencies as well. We consider the data from the ISI survey IT63c (20150206) and separate addresses that the whois 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 horizon-

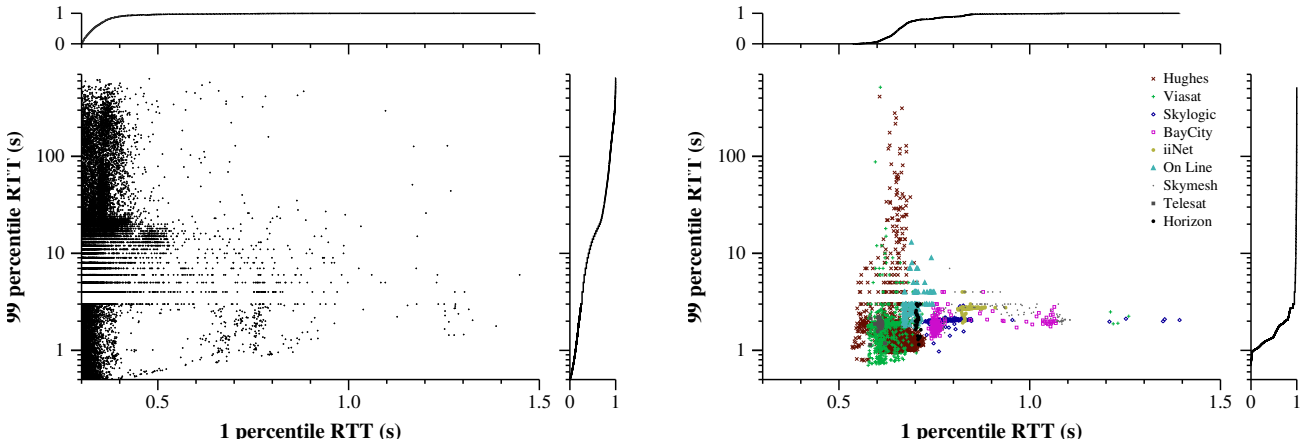


Figure 9: Scatterplot of 1st and 99th percentile latencies for addresses with high values of both in survey IT63c; Left omits satellite-only ISPs; Right includes only satellite-only ISPs.

tal bands.) 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 0.5 second first percentile.

Each 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 517 seconds—their 99th percentile values are predominantly below 3 seconds. They do not have such high 99th percentile values as the rest of the hosts with over 0.3 second first percentiles (those shown on the left graph).

6.2 Is it the first ping?

The latencies measured in Section 4 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.

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 10 the differences between the first and second round trip times for all those that had a first

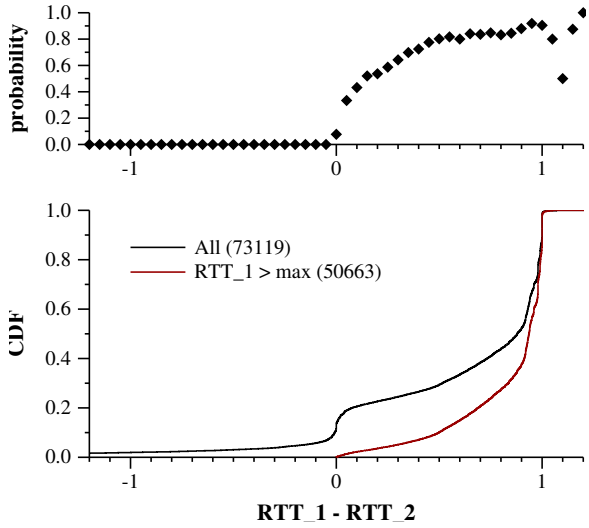


Figure 10: Bottom: Difference between initial latency and second probe latency; values around 1 indicate that both responses arrive at about the same time, values near zero indicate that the RTTs were about the same. The second line includes only those where $RTT_1 > max(RTT_2 \ldots RTT_n)$. Top: The probability that, given $RTT_1 - RTT_2$ on the x-axis, that $RTT_1 > max(RTT_2 \ldots RTT_n)$.

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.

We infer that a measurement approach that sent a second probe after one second could detect this behavior. The top graph of Figure 10 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 11 the differences between

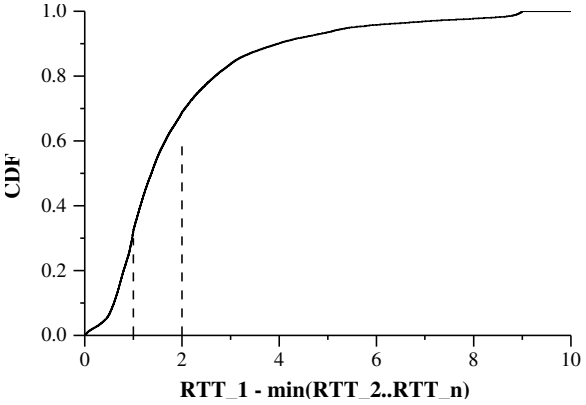


Figure 11: Difference between initial latency and observed minimum. The typical setup time is below four seconds.

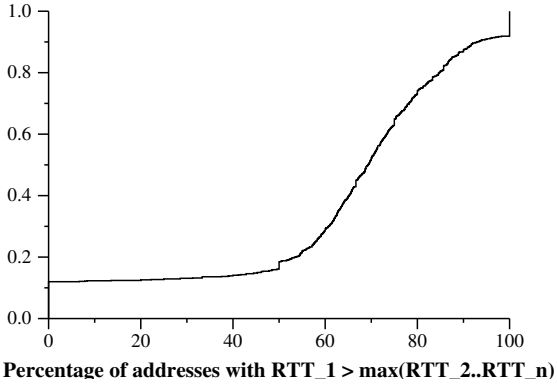


Figure 12: Percentage of addresses in a /24 prefix showing a drop from the initial to the maximum.

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 responsive addresses within each prefix that dropped from the

initial ping to the maximum of the rest in Figure 12. 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.

6.3 ASes with many high latency addresses

Next, we investigate high latency addresses in three datasets collected by Zmap in April 2015. Since Zmap sends an ICMP Echo Request to every IPv4 address, analyzing these responses allows us to recognize high-latency trends in Autonomous Systems and geographic locations.

Table 3 shows Autonomous Systems that are most prone to high latency. The table is sorted by the ASN with the most addresses that responded with a latency greater than 1 second to at least one of the three pings sent across the three Zmap scans.

The ASNs that are most prone to high latencies are cellular. Telefonica Brasil has the most addresses with latencies greater than a second; in fact, 85.1% of its addresses took more than 1 second to respond to at least one of the three Zmap pings. The next two ASNs, Bharti Airtel Ltd. AS for GPS Service, and Tim Celular, are also cellular, again with very high fractions of high latency addresses, and so are 5 of the remaining 7 ASes in the top 10. Chinanet and National Internet Backbone also feature in this list but their ratio of high latency addresses is low. We speculate that these ASNs offer cellular services in addition to other services, so that some of their addresses observe high latencies while others do not.

We next look at ASNs who responded with high latencies for all three pings sent to them for a large fraction of their addresses. This indicates ASNs whose addresses consistently see high latencies. Table 4 shows the organizations who had the highest fraction. We can confirm Skylogic, Bordernet and Iridium to be satellite providers. We speculate that Iren Energia is also a satellite provider.

We mapped addresses to their respective continents using MaxMind and show the continents with the most high latency addresses in Table 5. South America and Asia alone account for around 72% of all addresses which see latencies greater than 1 second. Further, more than a third of all addresses in South America and Africa experienced high latencies at least once. On the other hand, North America has only 2,216,033 addresses with high latencies. More than half of those addresses come from a single ASN (AS22394).

Continent	Responding Addresses	1^+ High RTT Addresses	%
South America	40459857	14120375	34.9
Asia	184835809	11038319	6.0
Europe	116682567	4776016	4.0
Africa	7170891	2614628	36.5
North America	106376285	2216033	2.1
Oceania	2765124	168564	6.1

Table 5: Continents with the largest number of addresses responding to Zmap probes after one second in at least one of the three scans.

6.4 What causes latencies above 100 seconds?

Next we look at addresses with extraordinarily high latencies (> 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 event, 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 one 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 a 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 by several responses of even greater latency, possibly with intermittent losses. This behavior is usually sustained for several minutes with

ASN	Owner	Responding Addresses	1+ High RTT Addresses	%
26599	TELEFONICA BRASIL	7924763	6741692	85.1
45609	Bharti Airtel AS for GPRS	2978143	2609728	87.6
26615	Tim Celular S.A.	2784045	2263161	81.3
22394	Cellco Partnership	1452063	1175136	80.9
9829	National Internet Backbone	3107603	1146755	36.9
4134	Chinanet	53061931	1117048	2.1
6306	VENEZOLAN	1274546	1115003	87.5
1257	TELE2	1319490	1051484	79.7
27831	Colombia Móvil	1331259	1039363	78.1
35819	Etihad Etisalat (Mobily)	1467583	1022709	69.7

Table 3: Autonomous Systems with the most addresses with latencies greater than 1 second in at least one of the 3 Zmap scans.

ASN	Owner	Responding Addresses	1+ High RTT Addresses	%	2+ High RTT Addresses	%	All High RTT Addresses	%
22184	IRIDIUM SATELLIT	30	25	83.333	21	70.000	17	56.667
39636	IREN ENERGIA S.P.A	3333	2754	82.628	2598	77.948	1774	53.225
23963	Bordernet Internet Pty Ltd	687	661	96.215	534	77.729	302	43.959
29286	SKYLOGIC S.P.A.	29844	26064	87.334	20229	67.782	10819	36.252

Table 4: Autonomous Systems with the highest ratio of all three addresses recording latencies > 1s over all their addresses. These are mostly satellite ASNs.

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

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 6. For each pattern, we show the number of pings > 150 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*.

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 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.

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 *Proceedings of the USENIX Symposium on Networked Systems Design & Implementation (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.