Time's Forgotten: Using NTP to understand Internet Latency

Ramakrishnan Durairajan[†], Sathiya Kumaran Mani[†], Joel Sommers[*], Paul Barford[†+]
{rkrish,sathiya,pb}@cs.wisc.edu, jsommers@colgate.edu
[†]University of Wisconsin - Madison [+]comScore, Inc. [*]Colgate University

ABSTRACT

The performance of Internet services is intrinsically tied to propagation delays between end points (*i.e.*, network latency). Standard active probe-based or passive host-based methods for measuring end-to-end latency are difficult to deploy at scale and typically offer limited precision and accuracy. In this paper, we investigate a novel but non-obvious source of latency measurement—logs from network time protocol (NTP) servers. Using NTP-derived data for studying latency is compelling due to NTP's pervasive use in the Internet and its inherent focus on accurate end-to-end delay estimation. We consider the efficacy of an NTP-based approach for studying propagation delays by analyzing logs collected from 10 NTP servers distributed across the United States. These logs include over 73M latency measurements to 7.4M worldwide clients (as indicated by unique IP addresses) collected over the period of one day. Our initial analysis of the general characteristics of propagation delays derived from the log data reveals that delay measurements from NTP must be carefully filtered in order to extract accurate results. We develop a filtering process that removes measurements that are likely to be inaccurate. After applying our filter to NTP measurements, we report on the scope and reach for US-based clients and the characteristics of the end-to-end latency for those clients.

Categories and Subject Descriptors: C.2.2 [Network Protocols]: Applications; C.2.3 [Network Operations]: Network monitoring

Keywords: NTP; Internet latency; Measurement

1. INTRODUCTION

Empirical measurement of the Internet informs the development and configuration of systems, protocols and services. One of the most fundamental characteristics that can be measured is the time taken for a packet to traverse one or more links between a sender and receiver, referred to as *delay* or *latency*. As link bandwidths have increased over the past decades, application performance has become more and more dominated by effects of end-to-end latencies [1]. As a result, understanding Internet's latency characteristics is increasingly important.

Developing a broad understanding of the latency characteristics is extremely challenging due to the Internet's scale and dynamics. Although the dominating contributor to latency in the wide area Internet is *propagation delay*, which is essentially static, queuing delays, node processing delays, and routing changes each affect observed latency and complicate analysis. Moreover, because of the vast and distributed nature of the Internet, empirical studies typically rely on probe-based methods for measuring one-way or round-trip latencies [2–4], making it difficult or impossible to identify specific contributing factors to overall latency.

Although there have been several efforts to deploy systems for continuous collection of Internet path latencies [5–7], such systems operate from a limited set of nodes from which probes are emitted and the measurements are inherently tied to the management policies of service providers. Accurate one-way measurements of latency are further complicated by the need for careful clock synchronization [8]. As a result, comprehensively characterizing latency has remained elusive. These limitations have prompted researchers to investigate crafty methods for measuring delays between two arbitrary hosts [4] and to develop techniques to *infer* latencies between arbitrary hosts via Internet coordinate and embedding systems [9–12].

In this paper, we propose using a previously untapped source of measurement data—the Network Time Protocol (NTP)—to improve our understanding of latency throughout the Internet. Our arguments for using NTP begin with the fact that it is used by clients throughout the Internet. To further illustrate scale, there are nearly 4,000 servers in the `ntp.org` server pool alone [13], and primary servers receive about 10,000 requests per second and nearly 1B requests per day [14]. Next, is the fact that the delay estimation is inherent in the NTP protocol: logs contain timestamps that specify when requests to servers were sent and when

HotNets '15 November 16–17 2015, Philadelphia, PA USA

the requests are received, (among other information), which provides a direct measurement of one-way latency. Finally, if NTP can be used, it would eliminate the need to manage and maintain dedicated latency measurement infrastructure and associated probe traffic. Our long-term goal is to better understand the dynamic nature of the Internet through comprehensive characterization of Internet latency based on analysis of protocol messages exchanged between NTP servers and clients and to leverage this information for operational decision making, *e.g.*, in traffic engineering. To the best of our knowledge, our study is the first to consider NTP protocol and log data for examining Internet latency.

We consider the efficacy of using NTP to understand latency by examining logs collected from 10 servers in the United States, including 3 primary (stratum-1) servers and 7 secondary (stratum-2) servers. Three of the servers provide NTP synchronization over IPv6, and one serves clients over IPv4 (one server is dual-stack). Overall, our raw log data include 73,837,719 latency measurements to 7,369,029 unique clients worldwide (as indicated by unique IP addresses) collected over the period of one day.

We develop and evaluate a filtering process designed to eliminate invalid and inaccurate latency samples from the raw log files. Our approach first attempts to detect whether an NTP client is well-synchronized with a server through analysis of the observed time differences between consecutive requests from a given client. For example, based on the NTP algorithms, if we observe a shift to a slower polling frequency, we infer that a client has synchronized with the server [15]. Surprisingly, we find that many clients poll at constant intervals and to these clients we apply a general filter to eliminate any clients with relatively high variability in latency measurements. From these filtered data, we focus on the minimum observed delay between a given client and server.

Our analysis reveals a wide range of observed latencies across all servers. For example, around 99% of US-based clients have latencies less than 100 milliseconds to the server with which they synchronize, compared to an earlier survey from 1999 which showed that 90% of clients had latencies below 100 milliseconds to their server [16]. We also observe a highly diverse client-base from a geographic perspective, especially for the secondary (stratum-2) servers. This diversity is much less pronounced for primary (stratum-1) servers because they are more tightly controlled, and for IPv6-based servers since they generally have a smaller set of clients they serve.

In summary, there are three contributions of this work. First, we identify and shed light on the use of NTP server log data for comprehensive analysis of Internet latency. Second, we develop an approach for filtering invalid latency measurements that leverages NTP's synchronization algorithm and analyze latency measurements derived from logs of 10 US-based NTP servers. The third and final contribution of the paper is a preliminary analysis of Internet's latency characters which shows that around 99% of observed minimum latency values between US-based clients and servers are less than 100 milliseconds, and clients are geographically well distributed and diverse.

2. OVERVIEW OF NTP

Maintaining a consistent notion of time among computer systems with independently running clocks is an interesting and old problem in computer networking and distributed systems. The most widely used protocol in the Internet for time synchronization is the Network Time Protocol (NTP). It is also one of the oldest networking protocols, having been initially established in RFC 958 in 1985 [17] based on the earlier Time Protocol [18] and ICMP timestamp [19] mechanisms and the even older Internet Clock Service from 1981 [20]. Early versions of NTP provided clock synchronization on the order of 10s to 100s of milliseconds, which considering network link bandwidths and processor clock speeds of computers at the time is quite an achievement.

NTP version 4 is the current recommended standard [21] and is largely backward compatible with prior versions of the protocol. NTP uses a hierarchical organization of servers and time sources. At the top-level, referred to as stratum 0, are high-precision time sources such as atomic clocks and GPS-based receivers. Servers connected to these high-quality sources are referred to as stratum 1 or *primary* servers. Stratum 2 or *secondary* servers connect to stratum 1 servers, *etc.*, all the way down to stratum 15. Servers may also peer with each other in order to provide redundancy at a given stratum. NTP clients may connect to servers at any level, but typically only connect to secondary servers and higher.

Commodity operating systems often ship with a default NTP server (or set of servers) configured (*e.g.*, `time.windows.com`, `time.apple.com`, `0.pool.ntp.org`), but any host can be reconfigured to use a different server. In order to compute a high-quality time estimate, it is common for clients to synchronize with more than one server. In fact, when clients perform a DNS lookup on one of the `ntp.org` "pool" servers, the authoritative DNS servers respond with multiple server IP addresses that are geographically close to the requesting client. The `ntp.org` web site also maintains lists of stratum 1 and stratum 2 servers that can be manually configured for use (some servers require permission from the server operator).

While many details of the protocol and algorithms used in NTP are beyond the scope of this paper, it is important to understand *some* of the key messages exchanged when clients poll servers, and when servers poll other servers. Four timestamps are generated as a result of a polling round: the time at which a polling request is sent (t_0), the time at which the request is received at the server (t_1), the time at which the response is sent by the server (t_2), and the time at which the response is received by the client (t_3). The round-trip delay is computed as $(t_3 - t_0) - (t_2 - t_1)$, and the one-way delay is assumed to be statistically one-half the RTT. Since our logs are captured at the server, we do not have access to t_3, thus we use the one-way delay $t_0 - t_1$ as the basis for our study. Moreover, the logs do not contain information regarding whether a client's clock is in "close" synchronization with the server. As a result, we developed a filtering process to identify and remove log entries that are likely to contain inaccurate latency samples as we discuss in §4.

3. NTP DATA

In this section, we describe characteristics of the data sets used in our study. We focus on the diversity of the client base and provide high-level statistics for the 10 different log files.

3.1 NTP Data Collection

To collect the NTP log data used in our study, we contacted several network administrators and explained our goals. Three administrators responded by providing datasets from a total of 10 NTP servers, including 4 commercial NTP servers in Utah, 2 private NTP servers in California, and 4 university campus NTP servers in Wisconsin. To facilitate network latency analysis, we developed a lightweight tool (about 700 lines of C code) to process and analyze the NTP server logs [1].

3.2 Basic Statistics

Table 1 summarizes the basic statistics from each of the NTP server logs and some of the key properties of these servers such as server stratum, IP version supported, number of measurements observed in the log file, number of clients, and the unique number of countries across which the clients are distributed. The selection includes 3 stratum-1 servers and 7 stratum-2 servers with a combination of both IPv4 and IPv6 support. These logs include a total of 73,837,719 latency measurements to 7,369,029 unique worldwide clients, as indicated by unique IP addresses, collected over a period of one day. From these 73.83M latency measurements, we filtered 48.86M measurements due to malformed headers, packet errors, missing timestamps, negative latency values, leaving us with about 25M latency measurements which form the basis of our analysis.

Table 1 highlights the fact that these servers provide time synchronization service to a huge diversity of clients. Nearly all stratum-2 servers have clients located in tens to hundreds of different countries. For example, there are clients from 218 unique countries disciplining their clock with the U2 stratum-2 NTP server. Many stratum-1 servers, such as C1 and W1, only offer access to a restricted set of clients, thus these servers only provide synchronization to US-based clients and servers. Although the U1 server operates at stratum-1, it does not restrict its base of clients. Rather unsurprisingly, the client diversity of IPv6-only servers is less than IPv4 servers.

Table 1. Summary of NTP server logs used in this study.

Server Location	Server ID	Server Stratum	IP version	Total measurements	Invalid measurements	Total unique clients	Total unique countries
Wisconsin	W1	1	v4	13,463	11,439	688	1
	W2	2	v4	6,769,429	6,035,742	1,652,615	105
	W3	2	v4	1,947,203	1,863,562	310,265	51
	W4	2	v4	1,967,262	1,699,014	144,920	89
Utah	U1	1	v4	2,463,041	1,097,816	148,529	186
	U2	2	v4	37,719,777	22,946,303	1,755,583	218
	U3	2	v6	13,935,717	8,861,346	2,462,419	54
	U4	2	v6	8,266	5,138	1,814	2
California	C1	1	v4/v6	13,561	3,640	127	1
	C2	2	v4	9,000,000	7,341,591	892,069	169

Figure 1. Client footprint of W3 NTP server.

To further illustrate client diversity, we graphically represent the distribution of hosts by geolocating the IP addresses using MaxMind's IP geolocation service [22]. Figure 1 shows the client footprint of the W3 NTP server (footprints from other logs had similar reach). This figure highlights the opportunity for understanding and analyzing latency characteristics across a broad cross-section of the Internet.

4. FILTERING NTP DATA

On initial analysis of the data, we observed that many of the clients were either *(1)* apparently starting up and sending a rapid series of requests, followed by a significant slowdown in polling, or *(2)* exhibiting a major shift in polling frequency, likely due to a major client-side clock adjustment or to a network path change. These observations, along with other sources, *e.g.*, [23, 24] suggest that many of the latency samples in NTP logs may be skewed and thus not suitable for latency analysis. Even if we examine only the minimum latencies across every individual client, there is no guarantee of accuracy of the measurement sample, since the logs contain no explicit indication of whether a particular client is in "good" synchronization with a server.

We developed an approach for filtering out inaccurate latency measurements which leverages the fact that the NTP synchronization algorithm causes a client's polling interval to change depending on how well it is synchronized to the servers. In NTP, this polling behavior is governed by its clock discipline algorithm [15]. Initially, a client can poll quite rapidly (*e.g.*, every 2 seconds). The polling rate then typically decreases as the algorithm indicates that it is in

---

[1] The tool is an extension of `netdissect.h` and `print-ntp.c` available from `https://github.com/the-tcpdump-group/tcpdump` and is available upon request.

good synchronization with a server. Over time, the frequency may increase and/or decrease, depending on network conditions and local clock drift. The maximum (rapid) and minimum (infrequent) rates at which a client can poll are configurable values, and default to 64 sec and 1024 sec, respectively. A client may briefly exceed its maximum polling rate on startup or during operation, again depending on configuration settings.

We take advantage of these polling behaviors in order to infer the stage at which a client has synchronized with a server. In our NTP log data, we observe four different variations, as follows. *(1)* Monotonically increasing polling values: all the polling values are increasing and at some point reach a set of constant values. In this case, we only use those latency samples corresponding to the most infrequent polling values as they suggest that a client is in good synchronization with the server. *(2)* Monotonically decreasing polling values: polling rate is initially constant then starts to decrease. Clients exhibiting this behavior do so in reaction to degraded synchronization due to network conditions or local clock drift. Similar to (1), we only extract latency samples from the constant polling period, prior to the increase in polling rate. *(3)* Constant polling values: no variation in polling rate is observed. In this situation, we have no way to infer the synchronization state of a client. While we could assume that a constant polling rate implies good synchronization, this is not a safe assumption due to the fact that a client may simply have configured identical minimum and maximum polling rates (which is apparently not uncommon [14]), or exhibit a simplistic polling behavior. For this situation we apply a general filtering approach. First, we eliminate all clients for which the difference between the maximum and minimum observed latency is greater than a fixed threshold. We choose 100ms as a threshold since we focus on US-based clients in this paper. Any clients with latencies beyond this threshold have either a very bad clock or may use an old implementation of NTP that less correctly (or too slowly) alters the polling interval. Next, for the remaining clients, we find the median and standard deviation of the latency values and remove all those samples that are not within one standard deviation from the median. *(4)* Varying (non-monotonic) polling values. For this situation, we select the longest set of samples during which polling values increase for a given client, and remove all other samples. The assumption with this approach is based on (1), that a period of monotonically increasing polling intervals implies an improvement in synchronization.

To provide perspective and toward the goal of assessing the effectiveness of our filtering approach, we sent 10 ICMP echo requests (using the standard `ping` tool) to 250 US-based clients from the W3 NTP server. The clients were selected randomly from a combination of clients whose latency samples were selected as well as rejected by our filtering process. Of these 250 clients, we received ICMP echo replies from 152 hosts. Figure 2 shows scatterplot comparisons of the minimum (Left), maximum (Center) and average (Right) NTP-derived latencies against corresponding RTT/2 values from the ping measurements. In each plot, data points accepted by our filtering approach are shown in green, and rejected data points are shown in red. In the plots, we first observe that there are no extreme outliers colored green, which indicates that we correctly filter out inaccurate samples. Of the accepted data points, 71.3% of the minimum NTP-derived latencies are within 5 milliseconds of the ping measurements, and 12.5% differ by more than 10 milliseconds. These differences may be due to network changes that occurred between the time at which our NTP logs were collected, and the time at which we were able to collect the active measurements. They may also be due to deficiencies in the filtering process, and we are currently considering how to include additional NTP-specific techniques in our filtering approach. Nonetheless, these results suggest that with appropriate filtering techniques applied, NTP server data can indeed provide a potentially rich source of accurate latency measurement data.

5. INTERNET LATENCY

In this section, we provide an analysis of the general characteristics of one-way latency as revealed through NTP log data.

Latency Characteristics. Figure 3 shows box-and-whiskers plots for worldwide clients *before* filtering (top) and US-only *after* filtering (bottom) that discipline their clocks to the 10 NTP servers we consider. The interquartile range with median is shown, along with minimum and maximum latency values. For the worldwide clients, we first observe that for the two stratum-1 servers (W1 and C1) that restrict which clients and servers may synchronize to them, their interquartile ranges are very tight. Although U1 is also a stratum-1 server, it does not restrict which hosts may synchronize to it and while the median latency between it and its clients is relatively low (less than 100 milliseconds), its interquartile range is quite large. Similarly, with several of the remaining stratum-2 servers (*e.g.*, W2, U3, U4, C2), their interquartile ranges are fairly large and the maximum observed latency extends to about 1 second. The U2, U3 and U4 servers have particularly large interquartile ranges, with *median* latencies of 175-300 milliseconds.

When considering only US-based clients (bottom plot of Figure 3) after filtering the latency distributions substantially shift toward lower values and the interquartile ranges shrink, and sometimes significantly so. For example, for the C2 server, the median changes from about 150 milliseconds when considering *all* clients, to around 30 milliseconds when only US-based clients are considered, and the 75th percentile shrinks from close to 600 milliseconds down to about 45 milliseconds. Similarly, this behavior can be observed in other stratum-2 servers (*e.g.*, U3, U4).

Latency Distribution. Tables 2, 3 and 4 show the distribution of minimum, maximum and average latency values

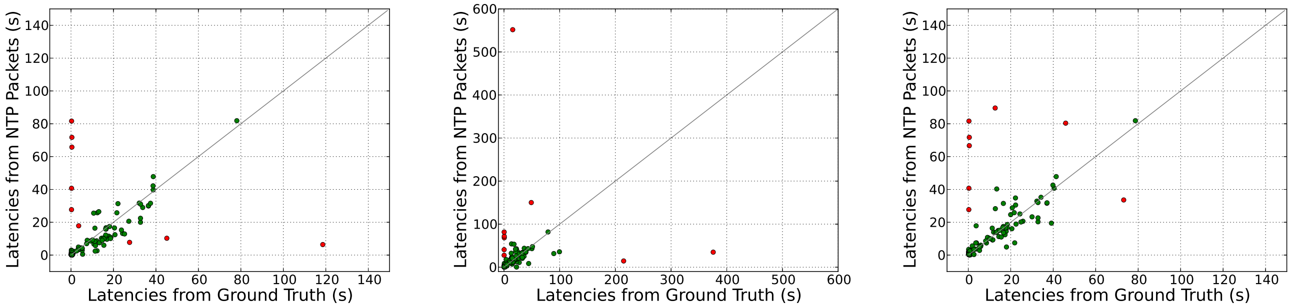

Figure 2. Comparison of measured minimum (Left), maximum (Center) and average (Right) latencies from NTP packets and ping measurements. Clients selected and rejected by the filter are denoted in green and red respectively.

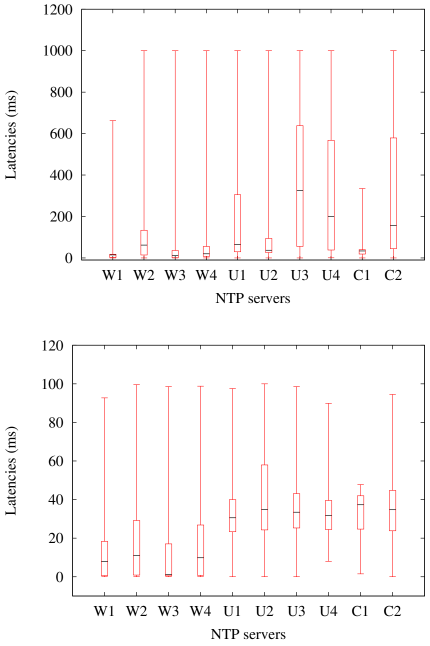

Figure 3. Box-and-whiskers plot showing latencies for clients distributed worldwide (top) before filtering and US-only (bottom) after filtering.

seen for US-only clients^2 (post-filtering). A survey [16] conducted in 1999 by Minar *et al.* (referenced in [25]) showed that *(i)* about 90% of the clients synchronizing their clocks with NTP servers had latencies below 100ms and about 99% were within one second, and *(ii)* stratum-1 servers were bottlenecked *i.e.*, they were serving too many clients and/or stratum-2 servers. For the servers considered in this study, and based on a much larger client base, we see somewhat different characteristics. First, many stratum-1 servers (including W1 and C1 in our study) restrict access to certain clients and stratum-2 servers as a way to ensure that they can accurately serve all their clients. This is different from what was observed in the survey. The effect on latencies is very clear both in the plots of Figure 3 and in these tables. For example, from the table, the minimum latencies observed between W1 and C1 and their US-only clients are all below 10 milliseconds, and while some maximum latencies are relatively high, only a small fraction of latencies are above 100 milliseconds.

Interestingly, although the median latencies are similar (within 10 milliseconds of each other) across servers deployed in a particular location (*e.g.*, across the 4 servers in Wisconsin, the 4 in Utah, and the 2 in California), the distributional characteristics vary significantly, depending on server load, stratum, and IP version supported. For example, for all the servers in Wisconsin, the majority of their clients have minimum latencies less than 10 milliseconds, in contrast with the Utah servers where the majority of the clients have minimum latencies between 20 to 30 milliseconds. Aggregating across all NTP servers for minimum, maximum, and average latency values, only 1%, 2.2% and 1.5% of the clients have latencies greater than 100ms. This observation differs significantly from the earlier survey of Minar [16] in which approximately 10% of clients had latencies above 100ms, and suggests that Internet latencies have improved since the time of that survey.

Table 2. Number of clients grouped in bins based on minimum latency values (in milliseconds) across all NTP servers.

Bins (ms)	W1	W2	W3	W4	U1	U2	U3	U4	C1	C2
0-10	24	3781	1366	1803	2707	19663	776	3	1	5173
10-20	5	1150	102	492	7775	30629	2296	23	1	4620
20-30	5	702	105	339	15486	56731	4169	60	1	7684
30-40	1	410	53	186	9834	33406	3831	56	6	11074
40-50	0	180	25	65	3975	19343	1406	26	2	6464
50-60	0	113	21	40	1753	12796	584	7	0	2490
60-70	0	86	41	32	1138	10685	228	1	0	1185
70-80	0	57	40	32	849	9673	90	0	0	822
80-90	0	41	42	20	680	9534	72	1	0	543
90-100	0	39	25	16	611	9280	47	0	0	473
>100	1	172	22	65	178	407	55	0	0	156

Client Counts and Locations. While Table 1 shows the number of world-wide unique clients and their locations as seen by the NTP servers before filtering, we show in Table 5 the number of unique US-only clients and their locations after filtering. From this table, we observe that the client base for each server is, in general, large and widely distributed. For example, for the U2 server, around 215K clients make requests from over 7900 cities across 48 states. Similarly,

---

[2] Clients can be wired or wireless. Characterizing how the increase in wireless clients has changed the latency distribution is part of the future work.

Table 3. Number of clients grouped in bins based on maximum latency values (in milliseconds) across all NTP servers.

Bins (ms)	W1	W2	W3	W4	U1	U2	U3	U4	C1	C2
0-10	18	2893	1215	1356	1235	9832	111	3	1	2689
10-20	10	910	134	422	6257	20777	876	15	1	4929
20-30	2	799	121	415	8290	36351	2327	55	1	5351
30-40	2	572	76	299	15767	42964	4125	54	1	10776
40-50	2	387	45	162	5866	28870	3019	36	7	8637
50-60	0	243	36	77	2775	18151	1647	11	0	3760
60-70	0	173	45	52	1516	13394	691	2	0	1661
70-80	1	188	44	50	1232	12109	261	0	0	1079
80-90	0	100	41	42	865	11020	154	1	0	768
90-100	0	75	34	34	833	11237	106	0	0	670
>100	1	391	51	181	350	995	237	0	0	364

Table 4. Number of clients grouped in bins based on average latency values (in milliseconds) across all NTP servers.

Bins (ms)	W1	W2	W3	W4	U1	U2	U3	U4	C1	C2
0-10	20	3133	1267	1490	1766	12245	413	3	1	3992
10-20	9	1118	135	500	6830	24853	1197	19	1	4564
20-30	3	818	118	432	12693	47405	3533	57	1	5575
30-40	1	547	74	237	12675	42141	4318	56	4	12321
40-50	1	297	33	117	4855	23051	2393	31	4	7571
50-60	1	205	27	59	2166	14552	894	9	0	2978
60-70	0	145	40	40	1313	11869	408	1	0	1307
70-80	0	91	40	47	1002	11015	130	0	0	955
80-90	0	65	39	33	739	10199	100	1	0	644
90-100	0	57	30	26	674	10077	60	0	0	525
>100	1	255	39	109	273	842	108	0	0	252

all other stratum-2 servers except W3 and U4 have clients spread across more than 40 states in the US. Interestingly, the v6-only U3 stratum-2 server has similar client-base characteristics as other high-traffic v4-only stratum-2 servers.

Table 5. Summary of unique US-only clients and their locations seen in the NTP logs.

Server ID	W1	W2	W3	W4	U1	U2	U3	U4	C1	C2
Unique clients	36	6731	1842	3090	44986	215819	13554	177	11	40684
Unique cities	18	1024	156	580	4677	7968	251	4	5	4531
Unique states	12	47	29	44	48	48	45	4	5	48

6. RELATED WORK

Measuring Internet path latencies has been a target of inquiry for decades. One of the earliest studies by Mills [26] used ICMP echo requests emitted from instrumented hosts ("Fuzzballs") to a few target systems to assess latencies. Since link bandwidths were quite low at that time, transmission delays played a much larger role than they do in the modern Internet. Moreover, since the full physical configuration of links was known and documented at this time, it somewhat easier to dissect causes behind observed latencies. Ten years after Mills's study, Bolot studied both Internet packet loss and latency, including queueing characteristics, in the Internet, still before its commercialization and greater decentralization [27].

Following commercialization of the Internet, it became much harder to comprehensively characterize *any* empirical property of the Internet, including latency. The highly influential work of Paxson in the mid-late 90s used a set of specially-deployed systems to carry out measurements of packet loss and latency [2]. This work informed much of the later work in this area, and the same systems deployed as part of his work were used a few years later to assess the *constancy* of certain Internet path properties such as loss, delay, and throughput [3]. There are a number of efforts today that take a similar approach of having specially-deployed systems to collect an essentially continuous stream of measurements such as latency, loss, and routing [5–7]. Besides empirical measurement of Internet latencies, there have been a variety of techniques developed to *estimate* latencies between arbitrary nodes in the Internet [4, 9, 28].

7. SUMMARY AND FUTURE WORK

Latency is an increasingly important factor behind the performance of many Internet protocols and services. In this paper, we propose the use of Network Time Protocol (NTP) server logs as an opportunistic source of accurate, Internet-wide latency measurements. We develop a data filtering technique designed to eliminate inaccurate latency samples that leverages details of NTP message exchanges, and experimentally show that our filter effectively removes poor data samples. We analyze latency characteristics of filtered data derived from 10 US-based NTP server logs. Our results show that about 99% of all latency samples from all US-based clients to all 10 servers are below 100 milliseconds, and that the client-base is quite large and highly geographically distributed.

In our ongoing work, we are refining our filtering algorithm and are pursuing a more comprehensive analysis of Internet latency characteristics by obtaining logs from additional servers, and through examination of a longer time scope with specific logs. We believe that using NTP as a window into developing a broader and deeper understanding of Internet latency is intrinsically valuable, and we intend to pursue two specific additional avenues of research. First, given that the client footprint of NTP servers is large and well-distributed, we hypothesize that it is possible to choose a set of NTP servers from which *all* core links in the Internet reside on some known physical path between a server and a corresponding client. For example, we might consider a constructive approach for identifying a set of NTP servers that can *cover* all physical network paths represented in the Internet Atlas database [29]. As a consequence, it may be possible to completely characterize latencies intrinsic to the physical links comprising the Internet. Second, we are also interested in reevaluating aspects of delay *constancy* [3] in such a comprehensive setting, as well as evaluating routing inefficiencies and delay inflation.

Acknowledgements

We thank Judah Levine (NIST) and Dave Plonka (Akamai) for their feedback and helpful discussions. We thank Tim Czerwonka (DoIT), James Babb (DoIT), Murray Anderegg (UNC), John Ricketts (Quintex) and Aaron Toponce for providing NTP server logs. This material is based upon work supported by the NSF under grant CNS-1054985, DHS grant BAA 11-01 and AFRL grant FA8750-12-2-0328. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF, DHS or AFRL.

References

[1] S. Sundaresan, W. De Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, “Broadband Internet Performance: A View from the Gateway,” in *ACM SIGCOMM*, 2011.

[2] V. E. Paxson, “Measurements and Analysis of End-to-end Internet Dynamics,” Ph.D. dissertation, University of California, Berkeley, 1997.

[3] Y. Zhang and N. Duffield, “On the Constancy of Internet Path Properties,” in *ACM IMW*, 2001.

[4] K. P. Gummadi, S. Saroiu, and S. D. Gribble, “King: Estimating latency between arbitrary internet end hosts,” in *ACM IMW*, 2002.

[5] “CAIDA’s Ark Project.” http://www.caida.org.

[6] “RIPE Atlas,” https://atlas.ripe.net, 2015.

[7] “PingER: Ping End-to-end Reporting,” http://www-iepm.slac.stanford.edu/pinger/, 2015.

[8] D. Veitch, S. Babu, and A. Pàsztor, “Robust Synchronization of Software Clocks across the Internet,” in *ACM IMC*, 2004.

[9] B. Wong, A. Slivkins, and E. G. Sirer, “Meridian: A Lightweight Network Location Service Without Virtual Coordinates,” in *ACM SIGCOMM*, 2005.

[10] T. E. Ng and H. Zhang, “Predicting Internet network distance with coordinates-based approaches,” in *IEEE INFOCOM*, 2002.

[11] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani, “A structural approach to latency prediction,” in *ACM IMC*, 2006.

[12] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: A Decentralized Network Coordinate System,” in *ACM SIGCOMM*, 2004.

[13] “NTP Pool Servers,” http://pool.ntp.org.

[14] J. Levine, “Personal Communications.” 2015.

[15] “NTP Polling Interval.” http://www.eecis.udel.edu/~mills/ntp/html/poll.html.

[16] N. Minar, “A Survey of the NTP Network,” December 1999. [Online]. Available: http://www.media.mit.edu/~nelson/research/ntp-survey99/

[17] D. Mills, “Network Time Protocol (NTP),” https://tools.ietf.org/html/rfc958, September 1985.

[18] J. Postel and K. Harrenstien, “Time Protocol,” https://tools.ietf.org/html/rfc868, May 1983.

[19] J. Postel, “Internet Control Message Protocol,” https://tools.ietf.org/html/rfc792, September 1981.

[20] D. Mills, “DCNet Internet Clock Service,” https://tools.ietf.org/html/rfc778, April 1981.

[21] D. Mills, J. Martin, J. Burbank, and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms Specification,” https://tools.ietf.org/html/rfc5905, June 2010.

[22] “MaxMind: IP Geolocation and Online Fraud Prevention,” https://www.maxmind.com/, 2015.

[23] “NTP Clock Filter Algorithm.” https://www.eecis.udel.edu/~mills/ntp/html/filter.html.

[24] J. Ridoux and D. Veitch, “Principles of Robust Timing over the Internet,” *Queue*, 2010.

[25] “NTP FAQ.” http://www.ntp.org/ntpfaq/NTP-s-algo.htm.

[26] D. Mills, “Internet Delay Experiments,” https://tools.ietf.org/html/rfc889, December 1983.

[27] J.-C. Bolot, “End-to-end Packet Delay and Loss Behavior in the Internet,” in *SIGCOMM CCR*, 1993.

[28] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, “IDMaps: A Global Internet Host Distance Estimation Service,” *IEEE/ACM TON*, 2001.

[29] R. Durairajan, S. Ghosh, X. Tang, P. Barford, and B. Eriksson, “Internet Atlas: A Geographic Database of the Internet,” in *ACM HotPlanet*, 2013.